Switch Engine™ v33.7.1 User Guide

Configuration, Commands, and Features

9041143-00 Rev AB
August 2026

Copyright © 2026 Extreme Networks

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: https://www.extremenetworks.com/about-extreme-networks/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: https://www.extremenetworks.com/support/policies/open-source-declaration/

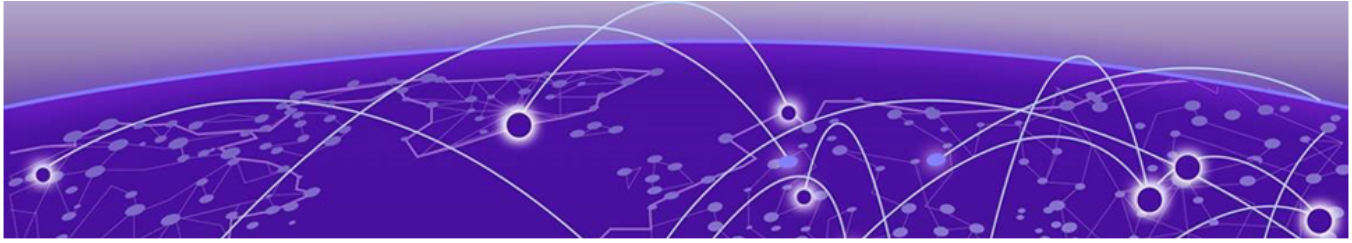


Table of Contents

Abstract vi

Introduction to the Switch Engine™ User Guide 7

Getting Started 9

Managing the Switch 67

Managing the ExtremeXOS Software 134

Configuring Stacked Switches 146

Configuring Slots and Ports on a Switch 223

ACLs 341

AVB 411

BGP 421

BGP Auto-peering 484

CLEAR-Flow 501

Data Center Solutions 520

EAPS 548

ERPS 604

ESRP 618

Ethernet Virtual Private Network (EVPN) 649

Extended Edge Switching 734

Fabric Attach 785

FDB 826

Identity Management 839

Integrated Application Hosting 868

IPv4 Unicast Routing 884

IPv6 Multicast 945

IPv6 Unicast Routing 955

IS-IS 991

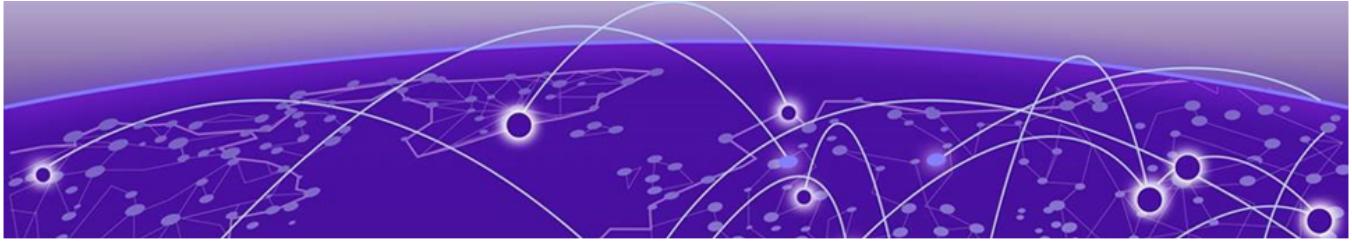
Keychain Manager 1013

Layer 2 Tunneling and Filtering 1016

Layer 3 Virtual Private Network 1024

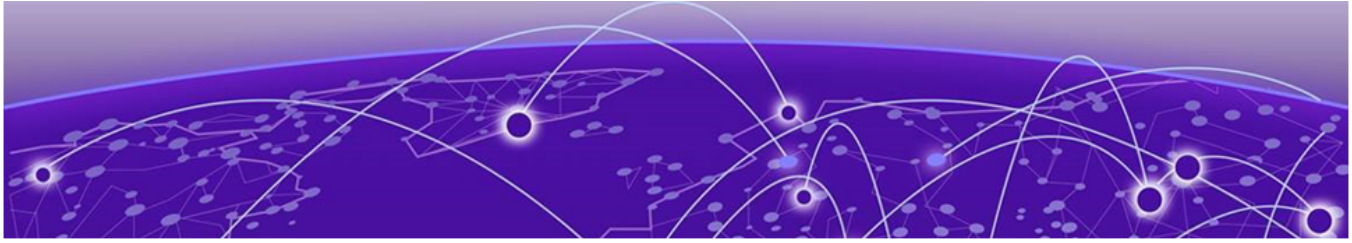
LLDP....1035
MPLS....1056
MSDP....1148
Multicast Routing and Switching....1157
Network Login....1214
OAM....1283
ONEPolicy....1318
OSPF....1391
OSPFv2 Auto-peering....1410
OSPFv3....1413
PoE....1428
Policy Manager....1448
Quality of Service....1453
RADIUS over TLS....1489
REST API....1496
RIP....1507
RIPng....1514
Routing Policies....1520
Security....1531
Service Probe....1686
Software Upgrade and Boot Options....1697
Status Monitoring and Statistics....1727
STP....1787
Supported Standards, Protocols, and MIBs....1854
Troubleshooting....2107
Universal Port....2135
Using CLI Scripting....2182
Virtual Routers....2198
VLANs....2213
VMAN (PBN)....2261
VRRP....2276
VXLAN....2305
Index....2363

Glossary..... 2341



Abstract

This user guide for Switch Engine version 33.7.1 details technical instructions for configuring and managing Extreme Networks switches, covering system setup, software prerequisites, stack management, VLAN configuration, CLI execution, Zero Touch Provisioning, configuration cloning, software upgrade procedures, and networking protocol implementation including BGP, OSPF, and VRRP. Administrators will find ExtremeCloud IQ agent onboarding and monitoring configuration procedures throughout. New features introduce automated RADIUS reauthentication pause using passive server reachability detection, which eliminates the need for manual CLI intervention or UPM scripts during a RADIUS outage, automatic session recovery for endpoints placed in a fallback VLAN during a RADIUS outage restoring full network access without administrator intervention once the server becomes available, on-demand Instant Port Profile rematch re-evaluating all matched devices against current switch state across all XIQ-supported platforms, and per-port Forward Error Correction statistics exposing corrected and uncorrectable block counters for Clause 74 and Clause 91/108 FEC modes to help diagnose marginal links, optics issues, and cabling problems, making this an essential reference for network administrators responsible for deploying and maintaining ExtremeSwitching hardware platforms.



Introduction to the Switch Engine™ User Guide

Send Feedback on page 7
Help and Support on page 8

This guide is intended for use by network administrators who are responsible for installing and setting up network equipment. In addition to comprehensive conceptual information about each feature of our software, you will also find detailed configuration material, helpful examples, and troubleshooting information. Also included are supported platforms and recommended best practices for optimal software performance.

Important
Release 31.6 introduced new names for the network operating systems running on Universal hardware. ExtremeXOS (EXOS) was renamed to Switch Engine and VSP Operating System Software (VOSS) was renamed to Fabric Engine. All references to ExtremeXOS apply to Switch Engine 31.6 and later.

Note
If the information in the release notes shipped with your switch differs from the information in this guide, follow the release notes.

Send Feedback

The User Enablement team at Extreme Networks has made every effort to ensure that this document is accurate, complete, and easy to use. We strive to improve our documentation to help you in your work, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.
- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, email us at Product-Documentation@extremenetworks.com.

Provide as much detail as possible including the publication title, topic heading, and page number (if applicable), along with your comments and suggestions for improvement.

Help and Support

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

The Hub

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Call GTAC

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2800. For the support phone number in your country, visit www.extremenetworks.com/support/contact.

Before contacting Extreme Networks for technical support, have the following information ready:

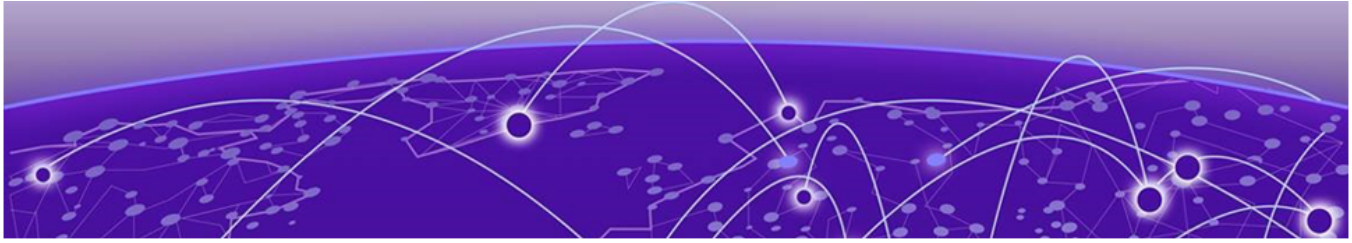
- Your Extreme Networks service contract number, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribe to Product Announcements

You can subscribe to email notifications for product and software release announcements, Field Notices, and Vulnerability Notices.

1. Go to The Hub.
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

You can modify your product selections or unsubscribe at any time.



Getting Started

Product Overview on page 10
Software Required on page 11
ExtremeCloud™ IQ Agent Support on page 12
Extreme Platform ONE Networking Licensing Support on page 15
Changing the Network Operating System on page 16
Simple Switch Configuration with Chalet on page 17
Switch Configuration Using VLAN ID (VID) on page 18
Zero Touch Provisioning (Auto Configuration) on page 18
Cloning Switches on page 23
Logging in to the Switch on page 31
Understanding the Command Syntax on page 32
Port Numbering on page 42
Line-Editing Keys on page 44
Viewing Command History (Journal) on page 45
Common Commands on page 45
Using Safe Defaults Mode on page 48
Configuring Management Access on page 50
Managing Passwords on page 56
Accessing an Active Node in a SummitStack on page 60
DNS Cache Resolver and Analytics Engine Overview on page 60
Domain Name Service Client Services on page 61
Checking Basic Connectivity on page 61
Displaying Switch Information on page 63
Filtering the Output of Show Commands on page 63
Automate Running Show Commands on page 65

The Getting Started chapter is intended to help you learn about your operating system software. Information about your product, software version requirements and navigation, common commands, and password management, along with other helpful software orientation information can be found in this chapter.

Product Overview

This table lists the Extreme Networks products that run the Switch Engine software.

Table 1: Switch Engine Switches

Switch Series	Switch Models
4120	4120-24MW-4Y 4120-48MW-4Y
4220	4220-8X 4220-12P-4X 4220-12T-4X 4220-24P-4X 4220-24T-4X 4220-48P-4X 4220-48T-4X 4220-4MW-8P-4X 4220-4MW-20P-4X 4220-8MW-40P-4X
5120	5120-24X-4Y 5120-24XT-4Y 5120-44X-4Y-2C
5320	5320-16P-2MXT-2X 5320-16P-4XE 5320-16P-4XE-DC 5320-24T-4X-XT 5320-24T-8XE 5320-24P-8XE 5320-24T-24S-4XE-XT 5320-48T-8XE 5320-48P-8XE

Table 1: Switch Engine Switches (continued)

Switch Series	Switch Models
5420	5420F-8W-16P-4XE 5420F-24P-4XE 5420F-24S-4XE 5420F-24T-4XE 5420F-16MW-32P-4XE 5420F-16W-32P-4XE 5420F-48P-4XE 5420F-48P-4XL 5420F-48T-4XE 5420M-24T-4YE 5420M-24W-4YE 5420M-16MW-32P-4YE 5420M-24W-24S-4YE 5420M-48T-4YE 5420M-48W-4YE
5520	5520-24T 5520-24W 5520-48T 5520-48W 5520-12MW-36W 5520-24X 5520-48SE 5520-24T-ACDC-BASE 5520-48T-ACDC-BASE 5520-24X-ACDC-BASE 5520-48SE-ACDC-BASE
5720	5720-24MW 5720-24MXW 5720-48MW 5720-48MXW
7520	7520-48Y-8C 7520-48XT-6C 7520-48YE-8CE
7720	7720-32C

Software Required

For information about which software version is required for each hardware switch model, see *ExtremeXOS and Switch Engine Software Support*.

The features available on each switch are determined by the installed feature license and optional feature packs. For more information, see the Switch Engine v33.6.1 Licensing Guide document.

ExtremeCloud™ IQ Agent Support

Switch Engine provides support for ExtremeCloud IQ. Currently, device discovery, basic monitoring, and visibility into homogenous stacking are supported. Switch Engine also has the ability to configure an optional user-defined virtual router (VR) and address of the server for ExtremeCloud IQ agent to connect to. These values are used instead of any auto-detected values.

In addition, users can configure the IQ Agent HTTP Proxy server IP and port, and define the username and password, if required.

To configure a server VR, VLAN Management, or address, use the following command:

```
configure iqagent server [vr [[vr-name | none] | [ vr_name vlan vlan-name]] | ipaddress [fqdn | ip_address| none]]
```

To configure the HTTP proxy, use the following command:

```
configure iqagent http-proxy [ipaddress [fqdn | ip_address] port port_number | user user_name password [encrypted encrypted_password | password] | none]
```

You can enable or disable the IQ Agent with the following commands:

Important
Disabling IQ Agent prevents all access to ExtremeCloud IQ. Any current activity with ExtremeCloud IQ, including remote SSH sessions, are disconnected immediately. Re-enabling IQ Agent can only occur by using the enable command by either console or Telnet or SSH access. Disabling IQ Agent deactivates automatic DHCP access on VLAN Mgmt, which is required for Zero-Touch Provisioning (ZTP).

```
enable iqagent

disable iqagent
```

To view information about IQ Agent, use the following command:

```
show iqagent discovery
```

For more information about ExtremeCloud IQ, go to https://www.extremenetworks.com/support/documentation/extremecloud-iq/.

Table 2: Supported Platforms

Switch Series	Switch Models
4120	4120-24MW-4Y 4120-48MW-4Y
4220	4220-8X 4220-12P-4X 4220-12T-4X 4220-24P-4X 4220-24T-4X 4220-48P-4X 4220-48T-4X 4220-4MW-8P-4X 4220-4MW-20P-4X 4220-8MW-40P-4X
5120	5120-24X-4Y 5120-24XT-4Y 5120-44X-4Y-2C
5320	5320-48T-8XE 5320-48P-8XE 5320-24T-8XE 5320-24P-8XE 5320-16P-2MXT-2X 5320-16P-4XE 5320-16P-4XE-DC 5320-24T-4X-XT 5320-24T-24S-4XE-XT
5420	5420F-8W-16P-4XE 5420F-24P-4XE 5420F-24S-4XE 5420F-24T-4XE 5420F-16MW-32P-4XE 5420F-16W-32P-4XE 5420F-48P-4XE 5420F-48P-4XL 5420F-48T-4XE 5420M-24T-4YE 5420M-24W-4YE 5420M-16MW-32P-4YE 5420M-24W-24S-4YE 5420M-48T-4YE 5420M-48W-4YE

Table 2: Supported Platforms (continued)

Switch Series	Switch Models
5520	5520-24T 5520-24W 5520-48T 5520-48W 5520-12MW-36W 5520-24X 5520-48SE 5520-24T-ACDC-BASE 5520-48T-ACDC-BASE 5520-24X-ACDC-BASE 5520-48SE-ACDC-BASE
5720	5720-24MW 5720-24MXW 5720-48MW 5720-48MXW
7520	7520-48Y-8C 7520-48XT-6C 7520-48YE-8CE
7720	7720-32C

IQ Agent and ExtremeCloud IQ Communication

After the IQ Agent is enabled, communication forms between the IQ Agent and extremecloudiq.com by secure HTTPS communication using destination TCP port 443. Communication between the IQ Agent and ExtremeCloud IQ occurs every 30 seconds, which includes check-in to ExtremeCloud IQ for actions, and includes CPU, memory, FDB information, Syslog, and ports statistics information. Note that data plane traffic is not sent to ExtremeCloud IQ by the IQ Agent.

Note that Telnet and SSH do not permit access to ‘hivemanager’ account, which the IQ Agent creates for its own purpose and uses it for all cloud-initiated SSH connections through local host, so logging on to this account through Telnet or SSH is not allowed.

IQ Agents use SNMPv2 (enabled only for internal requests) to monitor the status of the switch.

Distributed Denial of Service Support for IQ Agent

Distributed Denial of Service (DDoS) support for IQ Agent installs a filter on HTTPS L4 ports to set a CPU queue (QoS 5) that separates IQ Agent traffic from other IP exceptions. This new ACL redirects TCP traffic with source port 443 (default HTTPS port) to CPU queue 5. The IQ Agent system ACL is installed or uninstalled along with the L3 Unicast Miss (L3UCMiss) filter.

Automation of this feature is supported on all Universal switches.

You can also manually install the ACL to redirect IQ Agent traffic to CPU queue 5 on smaller switches with 8 ACL slices by running the following command:

```
# configure access-list iqagent.pol any

iqagent.pol:
entry iqagent_cpu5 {
  if {
    protocol tcp;
    source-port 443;
  } then {
    traffic-queue cpu_q_5;
  }
}
```

Extreme Platform ONE Networking Licensing Support

Extreme Platform ONE Networking is a unified touch point for Extreme Networks offered applications. It simplifies the user experience and provides automation at scale.

Extreme Platform ONE Networking includes three license levels: Standard, Advanced, and Premium. A Standard license is required to manage devices from ExtremeCloud IQ Agent.

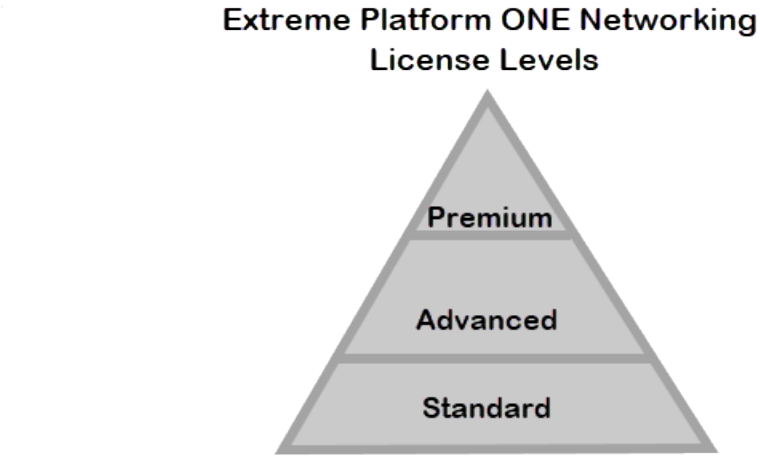


Figure 1: Extreme Platform ONE Networking License Levels

Each license level is purchased based on four tiers, depending on device type:

- A - 4000 series, 5120, 5320
- B - 5420
- C - 5520
- D - 5720, 7520, 7720

For full Universal licensing information, see Switch Engine v33.6.1 Licensing Guide .

Supported Platforms

All Universal platforms.

Changing the Network Operating System

The ExtremeSwitching Universal switches can run two different operating systems: Switch Engine (default) or Fabric Engine.

Making Your Initial Network Operating System Selection

You can make your initial selection of the operating system using:

- **ExtremeCloud IQ** (see ExtremeCloud IQ Agent Support)—You can select your network operating system when purchasing your switch, which associates the switch serial number with your desired network operating system, which then causes the desired network operating system to be loaded during ExtremeCloud onboarding. For more information about using ExtremeCloud IQ, go to https://www.extremenetworks.com/support/documentation/extremecloud-iq/.
- **Manually during boot-up**:
  - **Bootloader**—When the message `Starting Default Bootloader ...Press and hold the <spacebar> to enter the bootrom` appears, press and hold the **space bar** until the boot menu appears (you have 30 seconds):

```
*** 5520-24W Boot Menu ( 2.2.1.3 ) ***

     EXOS: Default
     EXOS: Primary 31.1.1.398
     EXOS: Secondary 31.1.1.398
     EXOS: Primary 31.1.1.398 with default configuration
     EXOS: Secondary 31.1.1.398 with default configuration
     EXOS: Rescue
     Change the switch OS to Fabric Engine
     Run Manufacturing Diagnostics
     Update bootloader
     Reboot system
```

    Use the **up** and **down** arrow keys to select `Change the switch OS to VOSS`, and then press **Enter**.
  - **Safe defaults mode start-up menu** (see Using Safe Defaults Mode on page 48—When the question `Would you like to change the switch OS to VOSS? [y/N/q]` appears:
    - For Switch Engine, type `N`.
    - For Fabric Engine, type `y`.

Continue to log onto the switch (see Logging in to the Switch on page 31).

Changing Your Network Operating System

You can change your network operating system selection at any time.

Caution
Changing your network operating systems deletes all configuration files, debug information, logs, events, and statistics information of the previous network operating system.

Note
If you anticipate ever changing the operating system to Fabric Engine, and you want to statically assign IP addresses on the DHCP server, then it is recommended to assign them based on the DHCP client ID. For more information about this issue, see Using a BOOTP or DHCP Server on page 73.

- **ExtremeCloud IQ**—See https://www.extremenetworks.com/support/documentation/extremecloud-iq/
- **CLI Command**—run the `download [url url {vr vrname} | image [active | inactive] [[hostname | ipaddress] filename {{vr} vrname} {block-size block_size}] {partition} {install {reboot}}` command specifying a VOSS image.

  Note
  Do *not* use the active, inactive, and partition options. They are not applicable for Fabric Engine.

Simple Switch Configuration with Chalet

Chalet is a web-based user interface for setting up and viewing information about a switch. Chalet removes the need to know and remember commands in a CLI environment. Viewable on desktop and mobile with a quick login and intuitive navigation, Chalet features an Quick Setup mode for configuring a switch in a few simple steps. Basic data surrounding port utilization, power, and Quality of Service (QoS) are available, and more advanced users can configure multiple VLANs, create Access Control Lists (ACLs), and configure Audio Video Bridging (AVB).

Chalet helps you interact with the switch outside of a CLI environment and allows you to easily:

- Configure the switch for the first time without the use of a console cable.
- View status and details of the switch and its slots and ports.
- Analyze power efficiency of power supplies, fans, and PoE ports.
- Create VLANs and ACL policies.
- Enable and disable multiple features, including QoS, AVB, auto-negotiation, and flooding.
- View recent system events.

- View device topology (stacked switches only).
- Manage users, including defining global and individual security policies.

Note
Beginning with version 31.4, access to the switch through Chalet is limited to only Admin users. Read-only users do not have access to the switch through the Chalet interface. This restriction is included due to security concerns.

For instructions about setting up, logging in, configuring, and monitoring your switch, see *Switch Configuration with Chalet for ExtremeXOS 21.x and 22.x*.

Switch Configuration Using VLAN ID (VID)

Referring to a *VLAN (Virtual LAN)* by VID and specifying lists of VIDs is a useful shortcut that can greatly reduce the number of commands required to configure the switch.

For example consider a scenario where two ports should be added to four tagged VLANs. Previously, this required four commands:

- configure vlan red add ports 2, 10 tagged
- configure vlan blue add ports 2, 10 tagged
- configure vlan green add ports 2, 10 tagged
- configure vlan orange add ports 2, 10 tagged

Assuming the tags for the VLANs are 100, 200, 300, 400 respectively this configuration can be accomplished with a single command:

- configure vlan 100,200,300,400 add ports 2, 10 tagged

Note
commands enhanced with VID list support operate in a “best effort” fashion. If one of the VIDs in a VID list do not exist the command is still executed for all of the VIDs in the list that do exist. No error or warning is displayed for the invalid VIDs unless all of the specified VIDs are in valid.

Zero Touch Provisioning (Auto Configuration)

Zero Touch Provisioning (ZTP) enables switches "just out of the box" to automatically gain a management IP address and configuration without serial cables and manual configuration. ZTP provides:

- Management port IP connectivity using an IPv4 link-local IP address.
- DHCP client to contact a *DHCP (Dynamic Host Configuration Protocol)* server for:
  - Assigned IP address
  - Switch Engine image update
  - Configuration or script file

IPv4 Link-Local Address

Link-Local addressing (subnet `169.254.x.x`) allows a host device to automatically and predictably derive a non-routable IP address for IP communication over Ethernet links.

By configuring the Ethernet management port "just out of the box" with an IP address, a user can connect a laptop directly to the management Ethernet port. If the laptop is not configured with a fixed IP address, it tries to get an IP address from a DHCP server. If it cannot, it assigns its own Link-Local address putting the switch and the laptop on the same subnet. The laptop can then use Telnet or a web browser to access the switch removing the need for the serial cable.

Note
Most 5320 models do not have dedicated management ports. You can use front panel ports for management connectivity for these switches. Models 5320-24T-4X-XT and 5320- 24T-24S-4XE-XT have dedicated management ports.

The IPv4 address format is used to make it simple for a user to determine the switch's IP address. The formula is to use the lower 2 bytes of the MAC address as the last two numbers in the Link-Local IPv4 address.

- MAC address: `00:04:96:97:E9:EE`
- Link-Local IP address: `169.254.233.238` or `0xa9fee9ee`

Web browsers accept a hexadecimal value as an IPv4 address. (Microsoft IE displays the URL with the number dot notation `169.254.233.239`.)

The web URL is `http://0xa9fee9ee` or just `0xa9fee9ee`

The user documentation directs the customer to access the web browser by typing `0xa9fe` followed by the last two number/letter groups in the MAC address found on the switch label. No hexadecimal translation is required.

With this information, you can connect the Ethernet port directly from a laptop to this switch using the temporary Link-Local address. You can communicate via web or Telnet to perform the initial switch configuration, if needed, and no longer needs a serial cable to configure a switch.

DHCP Parameters

If a DHCP server is available, ZTP tries to contact it alternating between the default *VLAN* and the management Ethernet port. The DHCP server can provide:

- IP Address
- Gateway
- Option 43 on page 20
- Options 66 and 67 on page 21

If an IP address is provided by a DHCP server on the management port, it replaces the Link-Local management IPv4 address.

If a TFTP server IP address is provided along with the name of a config file, ZTP downloads the config file to the switch. The switch reboots to activate the config file.

For .xos image files, ZTP executes the `download image` command to update the switch software. The switch does not reboot after the download image command completes.

DHCP Discovery Over Tagged VLANs

To discover Extreme Cloud, DNS queries must be carried over the tagged VLAN. On an unconfigured device, Auto Provisioning is enabled by default. As a result, the device must automatically discover the tagged VLAN DHCP service. During system initialization, an active agent starts and performs the following tasks:

1. Sends DHCP discover requests on all VLAN tags 2-4094.
2. Listens for any DHCP offer responses and records the VLAN and port.
3. For each VLAN that received a DHCP offer then:
   - Creates the VLAN in the operating system.
   - Adds the tagged port to the VLAN.
   - Attempts to resolve the following DNS names: `extremecontrol` and `hac.extremecloudiq.com`.
4. If neither domain name resolves, the system deletes the VLAN. If it finds at least one DHCP or DNS server, it starts the Cloud Connector (ZTP+) application.

Option 43

Option 43 processing does not require an NMS. If a switch receives option43 as part of the DHCP response, it uses the TFTP protocol to transfer files from the specified TFTP server IP address.

Option43 parameters may contain:

- TFTP Server to Contact
- Config file to be loaded or script to be run (.xsf or .py)
- Policy files (.pol)
- Switch Engine image file to be downloaded (.xos)
- Switch Engine XMOD file to be downloaded (.xmod)
- *SNMP (Simple Network Management Protocol)* trap receiver address for Extreme MIB traps

Multiple file names may be specified in option43. The file names can be either relative path names or a full URL with the IP address of the TFTP server. If relative path names are specified, the TFTP IP address is also required.

File name examples assuming a TFTP server is present with the IP address 10.10.10.1:

- `exos/summitX-15.7.1.1.xos` (specify the IP address in sub-option 100)
- `tftp://10.10.10.1/exos/summitX-15.7.1.1.xos` (sub-option 100 is not required)

Once all of the files specified in option43 have been transferred to the switch, the switch reboots.

Following is a sample Linux DHCP configuration:

```
option space EXTREME;
option EXTREME.tftp-server-ip code 100 = ip-address;
option EXTREME.config-file-name code 101 = text;
option EXTREME.snmp-trap-ip code 102 = ip-address;
class "Edge-without-POE" {
match if (option vendor-class-identifier = "XSummit");
vendor-option-space EXTREME;
option EXTREME.tftp-server-ip 10.120.89.80;
option EXTREME.config-file-name "XSummit_edge.cfg";
option EXTREME.snmp-trap-ip 10.120.91.89;
}
class "Edge-SummitX-POE" {
match if (option vendor-class-identifier = "XSummit");
vendor-option-space EXTREME;
option EXTREME.tftp-server-ip 10.120.89.80;
option EXTREME.config-file-name "xSummit_edge.xsf";
option EXTREME.snmp-trap-ip 10.120.91.89;
}
subnet 10.127.8.0  netmask 255.255.255.0 {
option routers                  10.127.8.254;
option domain-name-servers      10.127.8.1;
option subnet-mask              255.255.255.0;
pool {
deny dynamic bootp clients;
range 10.127.8.170 10.127.8.190;
allow members of "Edge-without-POE";
allow members of "Edge-SummitX-POE";
}
}
```

Options 66 and 67

Option 66 and option 67 provide TFTP server and `bootpfilename` for cases when option 43 is not available for ZTP.

Options 66 and 67 are received as DHCP options in a DHCP response by Switch Engine.

- Option 66 is used to identify the TFTP server with details of TFTP server IP address.
- Option 67 provides the `bootpfilename` details, which are downloaded to Switch Engine from the TFTP server IP address, and Switch Engine is rebooted after the download is successful. The `bootpfilename` can be of any image type (`.xos` or `.xmod`) or configuration file (`.xsf` or `.py`).

If option 43 is not present, then Switch Engine looks for the TFTP server IP address and bootp file name in options 66 and 67 to load the configurations or update the new image. If option 43, and options 66 and 67 are present, option 43 has higher precedence.

Onboarding to ExtremeCloud IQ or ExtremeCloud IQ Site Engine

As part of Auto Configuration, an unconfigured switch or stack can automatically connect to ExtremeCloud IQ (cloud-based) or ExtremeCloud IQ Site Engine (on-premises) for configuration and ongoing management—without requiring manual setup.

The DHCP service starts automatically on the following interfaces:

- Out-of-Band (OOB) Management port on the Mgmt VLAN
- Front Panel ports using either untagged (default) or tagged VLANs

ExtremeCloud IQ Agent and Cloud Connector

As part of auto configuration, ExtremeCloud IQ Agent and Cloud Connector start by default on an unconfigured device. After DHCP discovery completes and the DNS names `extremecontrol` (for ExtremeCloud IQ Site Engine) and/or `hac.extremecloudiq.com` (for ExtremeCloud IQ) are reachable, ExtremeCloud IQ Agent attempts to connect to ExtremeCloud IQ, and Cloud Connector attempts to connect to ExtremeCloud IQ Site Engine.

These agents use the virtual router VR-Default if the management servers are reachable through a tagged or untagged VLAN on a front panel port. If the servers are reachable through the OOB management port, they use VR-Mgmt instead. Communication between Cloud Connector and ExtremeCloud IQ Site Engine is also referred to as ZTP+.

Only one of the two—ExtremeCloud IQ or ExtremeCloud IQ Site Engine—is required to manage the device in a network.

Identifying an Auto Provision Enabled Switch

A switch enabled with auto-provisioning can be identified by its status, which appears in the output of the **`show auto-provision`** command.

Auto-bind on Spanning Tree Protocol Domain for Auto-provisioned VLANs

Version 32.5 modifies a Zero Touch Processing – Dynamic Host Configuration Protocol (ZTPDHCP) script to enable auto-bind on Spanning Tree Protocol Domain (STPD) "s0" for auto-provisioned Virtual Local Area Networks (VLAN). This allows the newly created VLAN(s) to participate in “s0” along with the default VLAN.

The ZTPDHCP script enables auto-bind by calling the following CLI command for every VLAN it creates:

`enable stpd` *`s0`* **`auto-bind`** `[ {`**`vlan`**`}` *`vid`*

If during ZTP the path to a DHCP server uses a tagged port, then the ZTPDHCP script auto-provisions a corresponding VLAN and adds the tagged port. This removes the port from STP Domain "s0", which may result in a network loop. Enabling auto-bind for the auto-provisioned VLAN on STPD "s0" provides loop protection.

The log file generated by the ZTPDHCP script logs the event whenever auto-bind is enabled on STPD “s0” for a newly created VLAN.

ZTPStack

The ZTPStack feature runs on the primary node of a stack and detects when new nodes join the stack topology. Once a new node is detected, ZTPStack configures its stacking parameters and reboots it.

If a node fails to join due to a software version mismatch, ZTPStack automatically copies the correct image to the node. It uses the cloning feature to update images across other nodes in the stack. Version mismatches typically occur when using features like push-button stacking or auto-stacking.

ZTPStack also supports Extended Edge Switching autoconfiguration.

For ZTPStack to function, stacking links must be active, which means stacking-support must be correctly configured or auto-discovery must be enabled. Auto-discovery configures stacking-support based on the transceiver and the capabilities of the connected device (refer to the *Auto-Discovery for Universal Hardware* section in the Configuring Stacked Switches chapter).

Cloning Switches

Switch Engine switches provide a script called `clone.py` that allows you to clone one switch's setup to another switch in the following scenarios:

- **Within a stack** (see Cloning within a Stack on page 24)
- **Standalone to standalone** (see Cloning Standalone to Standalone on page 26)
- **Standalone to stack** (see Cloning from Standalone to a Stack on page 28)
- **Standalone to USB** (see Copying One Switch's Configuration to Another Switch Using USB Zero Touch Provisioning (ZTP) on page 31)

The `clone.py` script copies the following:

- All content under:
  - `/boot`
  - `/exos`
  - `/alt/boot`
  - `/alt/exos`
  - (optionally) `/usr/local/cfg`
- Specific NVRAM contents:
  - Boot selector
  - CLI banner
  - Failsafe username,password, and action

Limitations

- Cloning to a standalone switch using a stacking master as the source does not make the standalone switch a stacking master. The configuration cloned from a stacking master to the standalone switch is ignored by the operating system.
- ONIE to non-ONIE or non-ONIE to ONIE cloning cannot be performed.
- Clone application does not connect through VR-USER.
- If the clone primary is started and stopped, the configuration dirty bit is set.

Supported Platforms

Cloning is supported on all ExtremeSwitching Universal platforms.

Using Clone.py

To run `clone.py` to copy the current switch, use the following command:

```
run script clone.py master
```

To stop `clone.py`, use the following command:

```
run script clone.py stop
```

To show if `clone.py` is running and display the version number, use the following command:

```
run script clone.py show
```

Cloning within a Stack

To clone within a stack, you can either:

- Run the `synchronize {slot slotid} {file_name}` command, which invokes the `clone.py` script. You can only clone one slot/switch at a time. For more information, see synchronize in the Switch Engine v33.6.1 Command References .
- Run the `clone.py` script directly, which allows you to specify multiple slots/switches. If more than one slot is specified, the master is cloned to each slot one at a time.

To clone within a stack by running the `clone.py` script directly, on the master switch, enter the command:

```
run script clone.py slot slotlist
```

Where *slotlist* can be any number of valid slot positions separated with commas, or all. For example, `run script clone.py slot 1,2,3`

The following appears on the master switch:

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                    C A U T I O N                     +
+   Cloning will erase all contents on these slots:   +
```

```
+                    1,3,4,5,6,7,8                    +
+     The switch contents of this stacking master     +
+     will replace the contents of these switches     +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++
Do you want to continue cloning? [y/N]: y
Cloning slot 1 started
.....................................
Slot 1 cloning COMPLETE
Cloning slot 3 started
.....................................................
Slot 3 cloning COMPLETE
Cloning slot 4 started
.....................................................
Slot 4 cloning COMPLETE
Cloning slot 5 started
.....................................................
Slot 5 cloning COMPLETE
Cloning slot 6 started
.....................................................
Slot 6 cloning COMPLETE
Cloning slot 7 started
.....................................................
Slot 7 cloning COMPLETE
Cloning slot 8 started
.............................................
Slot 8 cloning COMPLETE
```

On each slot, the following appears:

```
Cloning stack master. DO NOT REBOOT OR POWER CYCLE this switch
Starting cloning process
Transfering EXOS partitions and configuration

Transferring active EXOS partition.........
Updating active partition
.........................
Transferring inactive EXOS partition.................
Transferring /usr/local/cfg directory......
Transfering NVRAM information from master switch
Transfering control information from master switch

Rebooting to master switch configuration
You may need to adjust any IP addresses that may have been copied from the master switch
************************
*   Cloning COMPLETE   *
************************
Rebooting ...
reboot: Restarting system
```

Note

- The **synchronize {slot slotid}** command only works for cloning within the stack.
- To clone in a stack containing both 5720 and 5520 switches, you must download the image of the 5520 in a primary and secondary partition in the master slot.
- The **unconfigure switch all** command removes the summit-arm images that have been downloaded and saved in the cross stack.

Note

In mixed stacks containing both standard 5320 switches and 5320-16P-2MXT-2X switches, automatic image upgrades between model types are not supported. If a mismatch is detected, the system will display a warning and skip the upgrade. Similarly, when adding a 5320-16P-2MXT-2X switch to a mixed stack that includes 5520 or 5420 switches, automatic image upgrade for the 5320-16P-2MXT-2X is not supported.

To proceed, manually upgrade the image on the affected switch using the `download image` CLI command.

Note

In version 33.3.1 and earlier, the clone script used by `ZTPStack` and `sync slot` did not support downgrades from 33.4.1. To enable downgrade functionality:

- Load both the **ZTPStack XMOD** and **clone XMOD** onto the **Primary switch**.

Once loaded, both `ZTPStack` and `sync slot` will perform the downgrade correctly.

Cloning Standalone to Standalone

There are two common cases where you might want to clone from one switch to another:

Scenario 1: You want to have a reference ExtremeXOS image and configuration for your network. New switches can clone from the master switch, getting an exact copy as the beginning switch configuration that you can then modify as needed.

Scenario 2: You receive a large shipment of switches and want to configure one switch the way you want it, and then replicate that for all of the new switches. `Clone.py` is run on the clone master and the new switches can use *DHCP* to get a new address, and then copy from the clone master. The new switches must have an IP address and must be able to connect with the IP address of the clone master. Either a front panel port or the management port can be used. `Clone.py` tries both VR-Mgmt and VR-Default to connect with the IP address of the clone master.

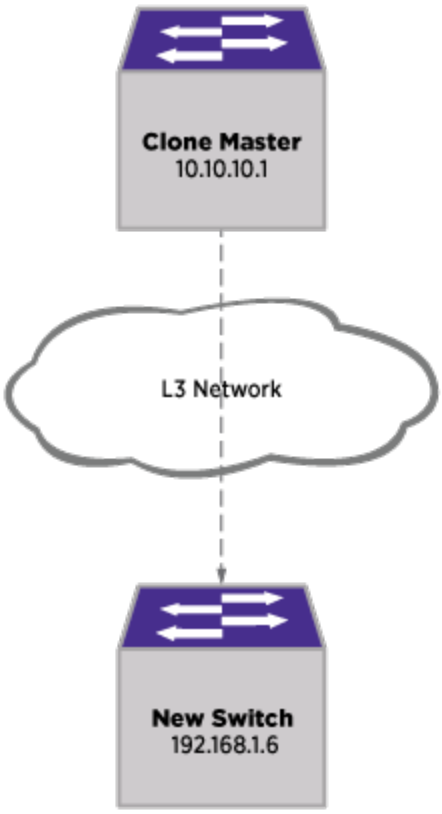


Figure 2: Cloning from Standalone Switch to Another Standalone Switch

To clone from standalone to standalone:

1. On the master switch, run the command `run script clone.py` **`master`**. The following appears:

```
# run script clone.py master
clone.py is started as cloning master
Use "stop" option to stop cloning master services
```

2. On the new switch (target), run the command `run script clone.py` **`from`** *`ipaddress`*. Where *`ipaddress`* is the IP address of the master switch. The following appears:

```
# run script clone.py from 10.68.65.81
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                             C A U T I O N                                  +
+      Cloning will replace the EXOS partitions and configuration            +
+   on this switch with the contents of the master switch 10.68.65.81   +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Do you want to continue cloning? [y/N]: y
clone.py is started as cloning client.
This switch will become a copy of 10.68.65.81
DO NOT REMOVE POWER or REBOOT the switch until the operation is COMPLETE

* (pacman debug) 5420F-48P-4XE.3 # Starting cloning process
Starting cloning process
Transfering EXOS partitions and configuration
```

```
Transferring active EXOS partition..............
Updating active partition
..............................................
Transferring inactive EXOS partition.....................
Transferring /usr/local/cfg directory..........
Transfering NVRAM information from master switch
Transfering control information from master switch

Rebooting to master switch configuration
You may need to adjust any IP addresses that may have been copied from the master
switch
************************
*   Cloning COMPLETE   *
************************
Rebooting ...
```

Cloning from Standalone to a Stack

You can easily add or replace a switch in a stack using the `clone.py` script.

When cloning to become a stacking member, additional NVRAM attributes are copied:

- Stacking enabled
- Stack MAC address
- Slot number—either provided or derived from the stacking master
- Switch's master-capable status—using the –m option when running `clone.py` sets the standalone switch to be master-capable.

Note
Cloning to a standalone switch using a stacking master as the source does not make the standalone switch a stacking master. The configuration cloned from a stacking master to the standalone switch is ignored by ExtremeXOS.

To clone from a standalone to a stack:

1. On the stack master, run the command `run script clone.py master`. The following appears:

```
clone.py is started as cloning master
Use "stop" option to stop cloning master services
```

2. Ensure that the standalone switch has an IP address or DHCP.
3. On the standalone switch, configure the switch's stacking-support options so that stacking connectivity is ready once the switch becomes stacking enabled. Run the command `enable stacking-support`.

Important
You must perform this step before cloning the standalone switch. If the switch is cloned as a stacking standby switch, console login and configuration options require contacting a stack master for user authentication. If the stacking links are not configured correctly, then only failsafe login is available.

4. On the standalone switch, confirm that the stacking links are configured correctly. Run the command `show stacking`. The following should appear (see bold text):

```
# show stacking
Stack Topology is a Ring
This node is not in an Active Topology
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:98:94:d3  -     Disabled     Master   --- <-This is the standalone switch
 00:04:96:98:87:3a  4     Active       Standby  --O
 0e:00:00:00:00:85  5     Active       Standby  --O
 00:04:96:98:87:54  6     Active       Standby  --O
 0e:00:00:00:00:84  7     Active       Standby  --O
 00:04:96:14:0b:03  8     Active       Backup   --O
 00:04:96:14:0b:04  1     Active       Standby  --O
 0e:00:00:00:00:83  2     Active       Master   --O
* - Indicates this node
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology
```

5. On the standalone switch, run the command `run script clone.py` **`from`** *`ipaddress`* **`-s`** *`STACKING_SLOT`***`-M`**, where:
   - *`ipaddress`* is the IP address of the stack master.
   - *`STACKING_SLOT`* is the slot number. Use 0 (zero) to have the switch given the lowest available slot number.
   - -M enable this switch to be master-capable.

   The following appears:

```
# run script clone.py from 10.68.69.94 -s 0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                              C A U T I O N                                   +
+         Cloning will replace the EXOS partitions and configuration           +
+     on this switch with the contents of the master switch 10.68.69.94        +
+                                                                              +
+                                  STACKING                                    +
+      You have requested this switch to be cloned to stack 10.68.69.94        +
+         Switches have a variety of stacking cabling options. See:            +
+                                                                              +
+                        configure stacking-support ...                        +
+                                  and                                         +
+                           enable stacking-support                            +
+                                                                              +
+   The stacking-support options MUST already be configured before cloning     +
+                  for stack connectivity to work correctly                    +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Do you want to continue cloning? [y/N]: y
clone.py is started as cloning client.
This switch will become a copy of 10.68.69.94
DO NOT REMOVE POWER or REBOOT the switch until the operation is COMPLETE

* # Starting cloning process
Starting cloning process
Transfering EXOS partitions and configuration

Transferring active EXOS partition.......
Updating active partition
................................
Transferring inactive EXOS partition..............
Transferring /usr/local/cfg directory....Transfering NVRAM information from master
switch
```

```
Transfering control information from master switch
Transfering stacking information from master switch
Using first available slot 3

Rebooting to master switch configuration
You may need to adjust any IP addresses that may have been copied from the master
switch
************************
*   Cloning COMPLETE   *
************************
Rebooting ...
reboot: Restarting system

Copyright 2014 Extreme Networks, Inc.

Starting CRC of Default image
Using Default image ...

Copyright 2014 Extreme Networks, Inc.


Press and hold the <spacebar> to enter the bootrom:  0
Loading Primary OS Image

Starting ExtremeXOS 22.2.0b38
Copyright (C) 1996-2017 Extreme Networks.  All rights reserved.
This product is protected by one or more US patents listed at http://
www.extremenetworks.com/patents along with their foreign counterparts.


 ______      _
|  ____|    | |
| |__  __  _| |_ _ __ ___ _ __ ___   ___
|  __| \ \/ / __| '__/ _ \ '_ ` _ \ / _ \
| |____ >  <| |_| | |  __/ | | | | |  __/
|______/_/\_\\__|_|  \___|_| |_| |_|\___|
 _   _      _                      _
| \ | |    | |                    | |
|  \| | ___| |___      _____  _ __| | _____
| . ` |/ _ \ __\ \ /\ / / _ \| '__| |/ / __|
| |\  |  __/ |_ \ V  V / (_) | |  |   <\__ \
|_| \_|\___|\__| \_/\_/ \___/|_|  |_|\_\___/

(pending-AAA) login:

Authentication Service (AAA) on the master node is now available for login.
admin
password:

ExtremeXOS
Copyright (C) 1996-2017 Extreme Networks. All rights reserved.
This product is protected by one or more US patents listed at http://
www.extremenetworks.com/patents along with their foreign counterparts.
================================================================================


You are connected to a Standby node.  Only a limited command set is supported.
You may use "telnet slot <slot_number>" to connect to the Master node to access
the full set of commands.

You are connected to a standby stack node's serial port.
Use "telnet slot <slot_number>" to connect to another node.

Press the <tab> or '?' key at any time for completions.
Remember to save your configuration changes.
```

```
* Slot-3 Stack.1 > show stacking
Stack Topology is a Ring
Active Topology is a Ring
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:98:94:d3  3     Active       Standby  CA-
 00:04:96:98:87:3a  4     Active       Standby  CA-
 0e:00:00:00:00:85  5     Active       Standby  CA-
 00:04:96:98:87:54  6     Active       Standby  CA-
 0e:00:00:00:00:84  7     Active       Standby  CA-
 00:04:96:14:0b:03  8     Active       Backup   CA-
 00:04:96:14:0b:04  1     Active       Standby  CA-
 0e:00:00:00:00:83  2     Active       Master   CA-
* - Indicates this node
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology
* Slot-3 Stack.2 >
```

Copying One Switch's Configuration to Another Switch Using USB Zero Touch Provisioning (ZTP)

You can copy one switch's configuration (ExtremeXOS image, configuration files, scripts) to another switch using USB Zero Touch Provisioning (ZTP).

Insert a USB drive with the desired configuration files on it into an unconfigured switch (does not need to be connected to the network), and the unconfigured switch automatically searches for the files and deploys them. The switch auto detects the USB drive at bootup or while ExtremeXOS is running.

Logging in to the Switch

Perform the following tasks to log in to the switch.

1. The initial login prompt appears as follows:

   `(Pending-AAA) login:`

   At this point, the failsafe account is now available, but the normal AAA login security is not. (For additional information on using the failsafe account, refer to Failsafe Accounts on page 55.)
2. Wait for the following message to appear:

   `Authentication Service (AAA) on the master node is now available for login.`

   At this point, the normal AAA login security is available.
3. Press **[Enter]**.

   Whether or not you press **[Enter]**, after you see the logon prompt, you can perform a normal logon. (See Default Accounts on page 54.)

   The following prompt appears: `login`

After you have successfully logged on, the following information about successful and unsuccessful logons appears:

```
There have been 26 successful logins since last reboot and 0 failed logins since last
successful login
Last Successful Login was on: Thu Apr 23 17:16:08 2020
```

Understanding the Command Syntax

This section describes the steps to take when you enter a command.

Software command syntax is described in detail in the Switch Engine v33.6.1 Command References . Some commands are also included in this guide in order to describe how to use OS software features. However, only a subset of commands are described here, and in some cases only a subset of the options that a command supports. You should consider the Switch Engine v33.6.1 Command References as the definitive source for information on commands.

You can enter configuration commands at the # prompt. At the > prompt, you can enter only monitoring commands, not configuration commands. When you log in as admin (which has read and write access), you see the # prompt. When you log in as user (which has only read access), you will see the > prompt. When the switch is booting up, you may see the > command prompt. When the bootup process is complete, the # prompt is displayed.

When you enter a command at the prompt, ensure that you have the appropriate privilege level.

Most configuration commands require you to have the administrator privilege level. For more information on setting CLI privilege levels, see the `create account`, `configure account`, and `configure account [all | name] privilege [admin | user]` commands.

Using the CLI

This section describes how to use the CLI to issue commands.

1. At the prompt, enter the command name.

   If the command does not include a parameter or values, skip to step 3. If the command requires more information, continue to step 2.
2. If the command includes a parameter, enter the parameter name and values.

   The value part of the command specifies how you want the parameter to be set. Values include numerics, strings, or addresses, depending on the parameter.
3. After entering the complete command, press **[Enter]**.

Note
If an asterisk (*) appears in front of the command line prompt, it indicates that you have pending configuration changes that have not been saved. For more information on saving configuration changes, see Software Upgrade and Boot Options on page 1697.

Syntax Helper

The CLI has a built-in syntax helper. If you are unsure of the complete syntax for a particular command, enter as much of the command as possible, and then press:

- **[Tab]**— Auto-completes the command if there is a unique match. If there is a partial match, auto-completes to the nearest match, and then lists the available options.
- **?**—provides a list of options for the entered command.

If you enter an invalid command, the syntax helper notifies you of your error, and indicates where the error is located.

If the command is one where the next option is a named component (such as a *VLAN*, access profile, or route map), the syntax helper also lists any currently configured names that might be used as the next option. In situations where this list is very long, the syntax helper lists only one line of names, followed by an ellipsis (...) to indicate that there are more names that can be displayed.

CLI File Name Completion

When entering a command, at the point where a file name could be entered, press **[Tab]** or **?** to display an alphabetically sorted list of possible file names. You can also type part of the file name to display a filtered list of file names matching what you have typed so far.

The following commands support this behavior:

- cd *directory_name*
- configure access-list *aclname* [**any** | **ports** *port_list* | **vlan** *vlan_name*] {**ingress**}
- configure ip-security dhcp-bindings storage filename *name*
- configure snmp access-profile [*access_profile* {**readonly** | **readwrite**} | [[**add** *rule*] [**first** | [[**before** | **after**] *previous_rule*]]] | **delete** *rule* | **none**]
- configure ssh2 access-profile [*access_profile* | [[**add** *rule*] [**first** | [[**before** | **after**] *previous_rule*]]] | **delete** *rule* | **none**]
- configure telnet access-profile [*access_profile* | [[**add** *rule*] [**first** | [[**before** | **after**] *previous_rule*]]] | **delete** *rule* | **none**]
- configure vlan *vlan_name* **udp-profile** [*profilename* | **none**]
- cp *old_name new_name*
- create process *name* executable *exe* {start [**auto** | **on-demand**]} {**node** *node*} {*vr vr-name*} {**description** *description*} {*arg1* {*arg2* { *arg3* { *arg4* { *arg5* { *arg6* { *arg7* { *arg8* { *arg9* }}}}}}}}}
- create process *name* **python-module** *python-module* {**start** [**auto** | **on-demand**]} {**node** *node*} {**vr** *vr-name*} {**description** *description*} {*arg1* {*arg2* {*arg3* {*arg4* {*arg5* {*arg6* {*arg7* {*arg8* {*arg9*}}}}}}}}}
- edit policy *filename*

- `enable ssh2 {access-profile [access_profile | none]} {port tcp_port_number} {vr [vr_name | all | default]}`
- `load script filename {arg1} {arg2} ... {arg9}`
- `ls file_name`
- `mkdir directory_name`
- `mv old_name new_name`
- `scp2 {cipher cipher} {mac mac} {compression [on | off]} {port portnum} {vr vr_name} user [hostname | ipaddress]:remote_file local_file`
- `show ssl {[trusted-ca | ocsp-signature-ca] [file_name | all]} {manufacturing}{certificate | detail}`
- TFTP
- `tftp get [ ip-address | host-name] { vr vr_name } { block-size block_size } remote-file local-file} {force-overwrite}`
- `tftp put [ ip-address | host-name] {vr vr_name} {block-size block_size}local-file { remote-file}`
- `rm file_name`
- `rmdir directory_name`
- `run script filename {arg1} {arg2} ... {arg9}`
- `unconfigure ssl certificate [trusted-ca | ocsp-signature-ca] [file_name | all ]`

Object Names

You must provide all named components within a category of the switch configuration (such as *VLAN*) a unique *object name*.

Object names must begin with an alphabetical character, and may contain alphanumeric characters and underscores (_), but they cannot contain spaces. The maximum allowed length for a name is 32 characters. User-created object names for the following modules are not case-sensitive: access list, account, *CFM (Connectivity Fault Management)*, *EAPS (Extreme Automatic Protection Switching)*, *ESRP (Extreme Standby Router Protocol)*, flow-redirect, meter, *MSDP (Multicast Source Discovery Protocol)*, Network Login, PVLAN, protocol, *SNMP*, SSHD2, STP, tunnel, UPM, *VLAN*, VMAN, etc.

Object names can be reused across categories (for example, *STPD (Spanning Tree Domain)* and VLAN names). If the software encounters any ambiguity in the components within your command, it generates a message requesting that you clarify the object you specified.

Note

If you use the same name across categories, we recommend that you specify the identifying keyword as well as the actual name. If you do not use the keyword, the system may return an error message.

Reserved Keywords

Keywords such as vlan, stp, and other second-level keywords are reserved and you cannot use them as object names. This restriction only applies to the specific word e.g (vlan); you can use expanded versions e.g (vlan2) of the word.

A complete list of the reserved keywords for ExtremeXOS 12.4.2 and later software is displayed in Table 3. Any keyword that is not on this list can be used as an object name.

Note

The list of reserved keywords is not exhaustive, and more will be added as new features are added. A failure to create an object with the following error means that the keyword is reserved:

"Name cannot be a reserved keyword."

Table 3: Reserved Keywords

Reserved Keywords				
aaa	elsm	irdp	pim	sys-health-check
access-list	ems	isid	poe	syslog
account	enable	isis	policy	sys-recovery-level
accounts	epm	jumbo-frame	ports	system
acl	erps	jumbo-frame-size	power	tacacs
auto-peering	esrp	kernel	printk	tech-support
auto-provision	ethernet	l2pt	private-vlan	telnet
avb	evpn	l2stats	process	telnetd
bandwidth	exit	l2vpn	protocol	tftpd
banner	exsh-var	lacp	put	throw
bfd	fabric	ldap	pwmib	thttpd
bgp	failsafe-account	learning	q	time
bootprelay	fdb	led	qosprofile	trusted-servers
bootrom	firmware	license	qosscheduler	tunnel
brm	flow-redirect	license-info	quit	twamp
bvlan	force_synchronize	licenses	radius	udp-echo-server
cdp	forwarding	licMgr	radius-accounting	udp-profile
cfgmgr	get	lldp	refresh	unconfigure
cfm	gptp	load	rip	update
checkpoint-data	hal	log	ripng	upload
clear	hclag	logout	router-discovery	upm
clear-flow	icmp	mac	rtmgr	usleep
cli	identity-management	mac-binding	run	validate
clipaging	idletimeout	mac-lockdown-timeout	provisioning	var
configuration	idmgr	mac-locking	save	version
configure	igmp	macsec	screen	vid
cos-index	image	management	script	virtual-network
counters	inline-power	mcast	security	virtual-router
cp	ip	mcm	session	vlan
cpu-monitoring	ip-fix	mcmgr	set	vm
create	ip-mtu	memory	sflow	vm-tracking
cvlan	ip-security	memorycard	sharing	vman
dcbx		meter	show	vpex

Table 3: Reserved Keywords (continued)

Reserved Keywords				
delete devmgr dhcp dhcp-client dhcp-server dhcpv6 diagnostics diffserv disable dns dns-client dos-protect dot1ag dot1p dot1q ds dwdm eaps edp elrp elrp-client	iparp ipconfig ipforwarding ipmc ipmcforwarding ipmroute iproute ipstats ipv6 ipagent	mirror mlag mld modify mpls mrp msdp msgsrv msrp mstp mv mvr mvrp neighbor-discovery netlogin nettools network-clock node nodealias ntp orchestration ospf ospvdebug ospfv3 otm packet performance	slot slot-poll-interval slpp snmp snmpMaster snmpv3 sntp-client source ssh2 sshd2 ssl stacking stacking-support stm stp stpd svlan switch	vr vrrp vsm watchdog web wred wredprofile xml-mode xml-notification xml-test xmlc xmld xx_force_synchronize xx_synchronize xx_synchronize2

Abbreviated Syntax

Abbreviated syntax is the shortest unambiguous allowable abbreviation of a command or parameter. Typically, this is the first three letters of the command.

When using abbreviated syntax, you must enter enough characters to make the command unambiguous and distinguishable to the switch. If you do not enter enough letters to allow the switch to determine which command you mean, the syntax helper provides a list of the options based on the portion of the command you have entered.

Command Shortcuts

Components are typically named using the `create` command. When you enter a command to configure a named component, you do not need to use the keyword of the component.

Note
True only for some named components.

For example, you can create a *VLAN*.

1. Enter a VLAN name:

   ```
   create vlan engineering
   ```

   After you have created the name for the VLAN, you can eliminate the keyword vlan from all other commands that require the name to be entered.
2. Enter the shortcut.

   ```
   configure engineering delete port 1:3,4:6
   ```

Command Aliases

You can create aliases to execute any ExtremeXOS command, including any options, arguments, and redirection.

For example, you can create an alias called "download" to substitute for "`download image 102.3.10.5`". Now you can now simply type "download" and then the image file name to download your ExtremeXOS image from the 102.3.10.5 location, instead of typing `download image 102.3.10.5 image_name`.

Important
Aliases are only available in the shell session in which they are created. When you exit the shell your aliases are lost. To create persistent aliases, you need to add the aliases to the script `exshrc.xsf` that you must create using the VI editor and save in the `/usr/local/cfg` folder.

For more information about the `exshrc.xsf` script, see ExtremeXOS Shell RC Script on page 2183. For information about managing files on the switch, see Using the ExtremeXOS File System on page 135.

To be recognized, the alias must be the first word in the command string typed at the prompt. Substitution does not occur if the alias name string occurs anywhere else.

Auto-completion works for aliases much as it does for ExtremeXOS commands (see Syntax Helper on page 33).

Limitations

- Arguments cannot occur in the middle of alias commands. For example, you cannot create an alias "set_vlan_ip" for the command `configure` **`vlan`** *`vlan_name`*

ipaddress *ip_address* where you specify the VLAN name as an argument. This is because aliases work through direct textual substitution.

- Aliases cannot be chained together. For example, if you create an alias "sh" for `show version` and another alias "ps" for `process`, then entering `sh ps` at the prompt is not equivalent to entering "`show version process`".
- You cannot tab-complete commands when creating an alias by using the `alias` command.
- Aliases cannot be created for the current shell session using UPM scripts or Python scripts.
- Tab completion does not work for commands given inside the aliases.

Usage

To create an alias, use the command `alias` *`alias_name command`*

To view a list of current aliases, use the command `alias` (with no other arguments).

To display the command that will be substituted for *`alias_name`*, use the command `alias` *`alias_name`*.

To delete aliases, use the command `unalias`.

Command Line Interface (CLI) History Expansion

The history expansion character '!' can be used to specify a command from the history that is substituted into the command line. All occurrences of the form "!n:w" in the command are replaced with the w'th word from the n'th line in the command history. Specification of the word is optional.

For example, if the command history is as follows:

1. show vlan
2. create vlan vlan_1
3. show ports

You can type `configure vlan !2:2 ipaddress 10.0.0.1` to configure the IP address of vlan_1. Or type `!1` to run the command `show vlan`.

You can also use negative values for n to refer backward through the history. `!-1` refers to the previous command executed, which in the preceding example is `show ports`. To execute a command that was typed two commands back, use `!-2`. You can use negative values for w to refer to the first (1+|w|) words from the selected line in history.

For example, `!2:-1` gives you the first 1+|-1|=2 words from line 2 of the history (in this example, `create vlan`).

This feature is disabled by default.

Supported Platforms

CLI history expansion is available on all ExtremeSwitching platforms.

Configuring CLI History Expansion

To enable CLI history expansion, use the following command:

`enable cli history expansion {session | permanent}`

To disable CLI history expansion, use the following command:

`disable cli history expansion {session | permanent}`

To view the status of CLI history expansion, use the following command:

`show switch management`

CLI History Lookup

You can find and re-execute CLI commands from the history.

To find a previously entered command:

1. Press **CTRL** + **R**. The prompt changes to `reverse-i-search`.
2. Start typing any part of the desired command. The CLI shows matching commands from the history. If there are multiple matches, press **CTRL** + **R** again to view additional matches.
3. To use the recovered command:
   - To execute the command with no changes, press **ENTER**.
   - To modify the command at the searched word, press **ESC** to place the command in the prompt with the cursor on the searched word.
   - To modify the command from the end of the command, press **CTRL** + **E** to place the command in the prompt with the cursor at the end of the command.

Support for Revising CLI Commands

ExtremeXOS has evolved and incorporated many new features over time. During this development, CLI keywords have been introduced that are not logically organized or do not conform to the CLI format standards. ExtremeXOS30.3 and later provides the infrastructure capability to manage the reorganization and removal of CLI commands with the goal of providing clearer CLI organization and improved usability.

The following command manages how old keywords that have been moved and redefined appear in the CLI:

`configure cli moved-keywords [hide | show {no-help}]`. Choices made in this command appear in the output of the `show management` command. The following

table explains the configure command options, how they impact CLI behavior, and how these choices appear in the `show switch management` command.

Moved Command Option	CLI Impact	Show Switch Management Command Display
hide	Moved commands do not appear in the CLI. • The removed keyword no longer appears as an option when using **TAB** or **?** for syntax suggestion and completion (see Syntax Helper on page 33). • Removed commands continue to run, provided the syntax is correct.	**Hidden**
show	Moved commands appear in the CLI. • The removed keyword appears as an option when using **TAB** or **?** for syntax suggestion and completion (see Syntax Helper on page 33). Help text indicates the new preferred syntax. • Removed commands continue to execute, provided the syntax is correct, but a message appears prompting you to use the preferred new syntax.	**Displayed with Help**
no-help	Moved commands appear in the CLI. • The removed keyword appears as an option when using **TAB** or **?** for syntax suggestion and completion (see Syntax Helper on page 33). Help text does not indicate the new preferred syntax. • Removed commands continue to execute, provided the syntax is correct. No message appears prompting you to use the preferred new syntax.	**Displayed with No Help**

Symbols

You may see a variety of symbols shown as part of the command syntax.

These symbols explain how to enter the command, and you do not type them as part of the command itself. Table 4 summarizes command syntax symbols you may see throughout this guide.

Note
ExtremeXOS software does not support the ampersand (&), left angle bracket (<), or right angle bracket (>), because they are reserved characters with special meaning in XML.

Table 4: Command Syntax Symbols

Symbol	Description
square brackets [ ]	Enclose a required value or list of required arguments. One or more values or arguments can be specified. For example, in the syntax `disable port [port_list | all]` you must specify either specific ports or all for all ports when entering the command. Do not type the square brackets.
vertical bar |	Separates mutually exclusive items in a list, one of which must be entered. For example, in the syntax `configure snmp add community [readonly | readwrite] alphanumeric_string` you must specify either the read or write community string in the command. Do not type the vertical bar.
braces { }	Enclose an optional value or a list of optional arguments. One or more values or arguments can be specified. For example, in the syntax `reboot {time month day year hour min sec } {cancel} {msm slot_id} {slot slot-number | node-address node-address | stack-topology {as-standby} }` You can specify either a particular date and time combination, or the keyword cancel to cancel a previously scheduled reboot. (In this command, if you do not specify an argument, the command will prompt, asking if you want to reboot the switch now.) Do not type the braces.

Port Numbering

ExtremeXOS runs on stand-alone and SummitStack switches, and the port numbering scheme is slightly different on each.

Stand-alone Switch Numerical Ranges

On switches, the port number is simply noted by the physical port number.

Separate the port numbers by a dash to enter a range of contiguous numbers, and separate the numbers by a comma to enter a range of non-contiguous numbers:

- x-y—Specifies a contiguous series of ports on a stand-alone switch.
- x,y—Specifies a non-contiguous series of ports on a stand-alone switch.
- x-y,a,d—Specifies a contiguous series of ports and a non-contiguous series of ports on a stand-alone switch.
- Port:Channel—For 4120, 5120-44X-4Y-2C, 5720, 7520, and 7720 channelized ports. For example, 49:4 maps to Port 49, Channel 4.

SummitStack Numerical Ranges

On SummitStack, the port number is a combination of the slot number and the port number. The nomenclature for the port number is as follows:

```
slot:port
```

For example, if the second switch in a SummitStack has four ports, the following ports are valid:

- 2:1
- 2:2
- 2:3
- 2:4

You can also use wildcard combinations (*) to specify multiple port combinations.

The following wildcard combinations are allowed:

- slot:*: Specifies all ports on a stack node.
- slot:x-slot:y: Specifies a contiguous series of ports on multiple stack nodes.
- slot:x-y: Specifies a contiguous series of ports on a particular stack node.
- slota:x-slotb:y: Specifies a contiguous series of ports that begin on one stack node and end on another stack node.

Assigning Slot:Port Notation on Standalone Switches

You can configure a standalone system as a slotted system, which allows for commands which had ‘slot’ arguments to be visible and take in a valid slot number of ‘1’, along with any port arguments specified in ‘slot’:’port’ notation. In turn, any command output would specify ‘slot’ information and ports displayed in ‘slot’:’port’ notation.

To assign slot:port notation to a standalone switch, use the following command:

```
configure system ports notation [slot:port | slot/port]
```

The CLI prompt for the switch changes to show that you have changed it to slot:port notation. For example:

```
Slot-1 5420F-48P-4XE.1
```

Port Notation on Standalone Channelized Ports

For 4120, 5120-44X-4Y-2C, 5720, and 7520 and 7720 series switches, each channelized port receives a specified channelized number. See the following table for examples using a 48-port switch model and 5720 VIMs.

Table 5: Switch Port Mapping

Switch channelized ports using the port option	Switch channelized ports using the slot:port option	5720 VIM-2CE channelized ports using the port option	5720 VIM-2CE channelized ports using the slot:port option
49:1	1:49:1	51:1	1:51:1
49:2	1:49:2	51:2	1:51:2
49:3	1:49:3	51:3	1:51:3
49:4	1:49:4	51:4	1:51:4
50:1	1:50:1	52:1	1:52:1
50:2	1:50:2	52:2	1:52:2
50:3	1:50:3	52:3	1:52:3
50:4	1:50:4	52:4	1:52:4

Note
5720 VIM-6YE ports are not channelized and map to ports 51-53 and 54-56.

Note
Switches running Release 31.6 or earlier that are connected to a channelized 5720 port will not display the correct port number via the Extreme Discovery Protocol. The port numbers will display correctly via the Link Layer Discovery Protocol.

Line-Editing Keys

Table 6 on page 44 describes the line-editing keys available using the CLI.

Table 6: Line-Editing Keys

Key(s)	Description
Left arrow or **[Ctrl] + B**	Moves the cursor one character to the left.
Right arrow or **[Ctrl] + F**	Moves the cursor one character to the right.
[Ctrl] + H or Backspace	Deletes character to left of cursor and shifts remainder of line to left.
[Ctrl] + D	Deletes character under cursor and shifts remainder of line to left.
[Ctrl] + K	Deletes characters from under cursor to end of line.

Table 6: Line-Editing Keys (continued)

Key(s)	Description
[Insert]	Toggles on and off. When toggled on, inserts text and shifts previous text to right.
[Ctrl] + A	Moves cursor to first character in line.
[Ctrl] + E	Moves cursor to last character in line.
[Ctrl] + L	Clears screen and moves cursor to beginning of line.
[Ctrl] + P or Up arrow	Displays previous command in command history buffer and places cursor at end of command.
[Ctrl] + N or Down arrow	Displays next command in command history buffer and places cursor at end of command.
[Ctrl] + U	Clears all characters typed from cursor to beginning of line.
[Ctrl] + W	Deletes previous word.
[Ctrl] + C	Interrupts the current CLI command execution.

Viewing Command History (Journal)

The operating system retains a list of the most recently executed CLI commands (journal). The journal retains as many as 200 commands along with the timestamp and user name. Commands are saved even after logging off, rebooting, or switch crashes.

To view the list of recent CLI commands, use the command `show cli journal`.

To configure command list size, use the command `configure cli` **`journal size`** *`size`*.

To view the currently configured command list size, use the command `show switch management`.

To remove the list, use either of the following commands: `unconfigure switch all` or `unconfigure switch erase all`.

Limitations

- Commands executed using North Bound interfaces (For example: *SNMP*, web) are not supported.
- The command `show cli journal` is not checkpointed in backup slot.

Common Commands

This section discusses common commands you can use to manage the switch.

Commands specific to a particular feature may also be described in other chapters of this guide. For a detailed description of the commands and their options, see the Switch Engine v33.6.1 Command References .

Table 7: Common Commands

Command	Description
`clear session [history | sessId | all]`	Terminates a Telnet or SSH2 session from the switch.
`configure account`	Configures a user account password. Passwords can have a minimum of 0 character and can have a maximum of 32 characters. Passwords are case-sensitive. User names are not case-sensitive.
`configure banner`	Configures the banner string. You can configure a banner to be displayed before login or after login.You can enter up to 24 rows of 79-column text that is displayed before the login prompt of each session.
`configure ports port_list {medium [copper | fiber]} auto off speed speed duplex [half | full]`	Manually configures the port speed and duplex setting of one or more ports on a switch.
`configure ssh2 key {pregenerated}`	Generates the SSH2 host key.
`configure sys-recovery-level [all | none]`	Configures a recovery option for instances where an exception occurs in the software.
`configure time month day year hour min sec`	Configures the system date and time. The format is as follows:`mm dd yyyy hh mm ss` The time uses a 24-hour clock format. You cannot set the year earlier than 2003 or past 2036.
`configure timezone`	Configures the time zone information to the configured offset from GMT time. The format of **`GMT_offset`** is ± minutes from GMT time. The **`autodst`** and **`noautodst`** options enable and disable automatic Daylight Saving Time change based on the North American standard. Additional options are described in the Switch Engine v33.6.1 Command References
`configure [{ vlan} vlan_name | vlan vlan_id ] ipaddress [ipaddress {ipNetmask} | ipv6-link-local | {eui64} ipv6_address_mask]`	Configures an IP address and subnet mask for a *VLAN*.

Table 7: Common Commands (continued)

Command	Description
`create account`	Creates a user account. This command is available to admin-level users and to users with *RADIUS (Remote Authentication Dial In User Service)* command authorization. The username is between 1 and 32 characters and is not case-sensitive. The password is between 0 and 32 characters and is case-sensitive. For user names, only alphanumeric, dash (-), and underscore (_) characters may be used. Any text after a hashtag (#) in a password is ignored. It has to be enclosed in "<double quotes>" if the hashtag is to be made part of the special character password.
`create vlan [{ vlan} vlan_name | vlan vlan_list ] {description vlan-description} {vr name}`	Creates a VLAN.
`delete account name`	Deletes a user account.
`delete [{ vlan} vlan_name | vlan vlan_list]`	Deletes a VLAN.
`disable bootp vlan [vlan | all]`	Disables BOOTP for one or more VLANs.
`disable cli prompting`	Disables CLI prompting for the session.
`disable cli config-logging`	Disables logging of CLI commands to the Syslog.
`disable cli paging`	Disables pausing of the screen display when a show command output reaches the end of the page.
`disable cli idle-timeout`	Disables the timer that disconnects all sessions. After being disabled, console sessions remain open until the switch is rebooted or until you log off. Telnet sessions remain open until you close the Telnet client. SSH2 sessions time out after 61 minutes of inactivity.
`disable port [port_list | all]`	Disables one or more ports on the switch.
`disable ssh2`	Disables SSH2 access to the switch.
`disable telnet`	Disables Telnet access to the switch.
`enable bootp vlan [vlan | all]`	Enables BOOTP for one or more VLANs.
`enable cli config-logging`	Enables the logging of CLI configuration commands to the Syslog for auditing purposes. The default setting is disabled.
`enable cli paging`	Enables pausing of the screen display when show command output reaches the end of the page. The default setting is enabled.

Table 7: Common Commands (continued)

Command	Description
`enable cli idle-timeout`	Enables a timer that disconnects all sessions (Telnet, SSH2, and console) after 20 minutes of inactivity. The default setting is enabled.
`install license` **`file`** *`filename`* {**`slot`** *`slot`*}	Installs a license key file on 5420 and 5520 series switches. This command installs all of the license features contained in the selected license file to the specified slot (node).
`enable ssh2` {**`access-profile`** [*`access_profile`* | **`none`**]} {**`port`** *`tcp_port_number`*} {**`vr`** [*`vr_name`* | **`all`** | **`default`**]}	Enables SSH2 sessions. By default, SSH2 is disabled. When enabled, SSH2 uses TCP port number 22.
`enable telnet`	Enables Telnet access to the switch. By default, Telnet uses TCP port number 23.
`history`	Displays the commands entered on the switch.
`show banner` {**`after-login`** | **`before-login`**}	Displays the user-configured banner.
`unconfigure switch` {**`all`**} or `unconfigure switch erase` **`all`** | **`nvram`**	Resets all switch parameters (with the exception of defined user accounts, and date and time information) to the factory defaults. If you specify the keyword **`all`**, the switch erases the currently selected configuration image in flash memory and reboots. As a result, all parameters are reset to default settings.

Using Safe Defaults Mode

When you take your switch from the box and set it up for the first time, you set the safe defaults mode. You should use the safe defaults mode, which lets you configure IP management connectivity if you choose not to use Auto Provisioning It also lets you choose whether to enable or disable Telnet, *STP (Spanning Tree Protocol)*, and *SNMP*. All ports are enabled in the factory default setting; you can choose to have all unconfigured ports disabled on reboot using the interactive questions. Also, *STPD* s0 is enabled on the default *VLAN*; you have the option to disable STPD in safe defaults mode.

After you connect to the console port of the switch, or after you run `unconfigure switch {all}` or `run provisioning`, you can change management access to your device to enhance security.

1. Connect the console and log in to the switch.

```
This switch currently has some management methods enabled for convenience reasons.
Please answer these questions about the security settings you would like to use.
You may quit and accept the default settings by entering 'q' at any time.

Would you like to change the switch OS to Fabric Engine? [y/N/q]
```

2. You are prompted to select your network operating system: Switch Engine (default) or Fabric Engine:
   - For Switch Engine, type `N`.
   - For Fabric Engine, type `y`.
3. Type `y` (to disable) or `n` (to enable) auto-provisioning.

```
By default, Auto-Provisioning uses DHCP on all Ethernet ports as this switch
attempts to connect to an Extreme Networks management product.
Instead of using DHCP, do you want to 'disable auto-provision' and
configure a static IP address, default gateway and DNS server now? [y/N/q]: y
```

   Select `y` to be prompted for a port to use for management, an IP address and subnet mask length, an optional default gateway IP address, a DNS name server and DNS domain suffix.
4. Type `y` (to disable) or `n` (to enable) *MSTP (Multiple Spanning Tree Protocol)*.

```
The switch offers an enhanced security mode. Would you like to read more,
and have the choice to enable this enhanced security mode? [y/N/q]:
```

   If you select "no," go to 6 on page 49.
5. If you select "yes," the following appears:

```
Enhanced security mode configures the following defaults:

        * Disable Telnet server.
        * Disable HTTP server.
        * Disable SNMP server.
        * Remove all factory default login accounts.
        * Force creation of a new admin (read-write) account.
        * Lockout accounts for 5 minutes after 3 consecutive login failures.
        * Plaintext password entry will not be allowed.
        * Generate an event when the logging memory buffer exceeds 90% of capacity.
        * Only admin privilege accounts are permitted to run "show log".
        * Only admin privilege accounts are permitted to run "show diagnostics".

Would you like to use this enhanced security mode? [Y/n/q]:
```

   If you select "yes," enhanced security mode is enabled. Go to step 10 on page 50.
6. If you select "no," you are prompted to disable Telnet:

```
Telnet is enabled by default. Telnet is unencrypted and has been the target of
security exploits in the past.

Would you like to disable Telnet? [y/N/q]:
```

7. You are prompted to enable SNMPv2c:

```
SNMP access is disabled by default.
SNMPv2c uses no encryption, SNMPv3 can be configured to eliminate this problem.
Would you like to enable SNMPv2c? [y/N/q]: Yes
```

8. You are prompted to set up the community string:

```
SNMP community string is a text string that is used to authenticate SNMPv2c messages.
It is required for managing the switch using SNMPv2c.
Would you like to configure a read-only and read-write community string? [Y/n/q]: Yes

Read-only community string:
Re-enter read-only community string:
Read-write community string:
Re-enter read-write community string:
```

9. You are prompted to enable SNMPv3:

```
Would you like to enable SNMPv3? [y/N/q]: Yes

SNMPv3 uses usernames/passwords to authenticate and encrypt SNMP messages.
Would you like to create an SNMPv3 user? [Y/n/q]: Yes

User name: admin
Authentication password:
Reenter authentication password:
Privacy password:
Reenter privacy password:

SNMPv3 user 'admin' was created with authentication protocol SHA and privacy protocol
AES-128.
```

10. Reboot the switch.

Configuring Management Access

Account Access Levels

By default, the software supports two roles for login sessions: administrative (or “admin” privilege having "write" permission) and non-administrative (or “user” privilege having "read-only" permission).

To change a user's account-level privileges to be administrative or non-administrative, use the command `configure account [all | name] privilege [admin | user]`. Any sessions that are currently logged in with the account specified are cleared and therefore forced to login again with the new privilege.

In addition to the account-level privileges, you can optionally use an external *RADIUS* server to provide CLI command authorization checking for each command. For more information on RADIUS, see Security on page 1628.

Note
CLI commands are sent to the RADIUS server unencrypted. Sensitive information entered into the CLI could be seen by either internal or external third parties.

User Accounts

A user-level account has viewing access to all manageable parameters. Users cannot access:

- User account database
- *SNMP* community strings

An account with a user-level privilege can use the `ping` command to test device reachability and change the password assigned to the account name.

If you have logged on with user privileges, the command line prompt ends with a (>) sign. For example: `5420-24T.3 >`

Administrator Accounts

Administrator-level accounts can view and change all switch parameters.

With this privilege level, you can also add and delete users, as well as change the password associated with any account name. To erase the password, use the `unconfigure switch all` command.

An administrator can disconnect a management session that has been established by way of a Telnet connection. If this occurs, the user logged on through the Telnet connection is notified that the session has been terminated.

If you log on with administrator privileges, the command line prompt ends with a pound or hash (#) sign.

For example: `5420F-48P-4XE.3 #`

Lawful Intercept Account

The Lawful Intercept account can log in to a session and execute lawful intercept commands on the switch. The commands provide for configuration consists of dynamic ACLs and a mirror-to port to direct traffic to a separate device for analysis. The lawful intercept login session, session-related events, and the ACLs and mirror instance are not visible to, or modifiable by, any other user (administrative or otherwise).

No lawful intercept configuration is saved in the configuration file, and it must be reconfigured in the case of a system reboot.

Other important feature information:

- An administrative user can create and delete a single local account having the lawful intercept privilege and user privileges, but not administrative privileges, and can set its initial password.
- The lawful intercept user is required to change the password (for the single lawful intercept-privileged account) upon logging in for the first time.
- The password for the lawful intercept account can only be changed by the lawful intercept user and cannot be changed by an administrative user.
- The `show accounts` command displays the existence of the lawful intercept account, but does not display any related statistics.

- The `show configuration` command does not display the lawful intercept account.
- The `show session {{`**`detail`**`} {`*`sessID`*`}} {`**`history`**`}` command does not display any lawful intercept user information. The EMS events normally associated with logging in and out are suppressed, and do not occur relative to logging in and out of the lawful intercept account.
- The EMS events normally associated with the `enable cli config-logging` command are suppressed, and do not occur relative to a lawful intercept user session.
- The lawful intercept user can create and delete non-permanent dynamic ACLs with the mirror action only. The lawful intercept user cannot create or delete any other ACLs.
- The `show access-list` command does not display any Lawful Intercept user-created ACLs to a non-lawful intercept user.
- The lawful intercept user-created ACLs are not accessible for any use by a non-lawful intercept user (specifically through the `configure access-list add` or `configure access-list delete` commands).
- The lawful intercept user can only create or delete one (non-permanent) mirror instance with which to bind the lawful intercept user-created ACLs and specify the mirror-to port.

Configure Banners

You can add a banner to give users helpful information before or after logging in.

- To add a banner to your switch, use the following command:
  `configure banner {`**`after-login`** `| { `**`before-login`** `} { `**`acknowledge`** `} | `**`before-login`** `{`**`acknowledge`**`} `**`save-to-configuration`**`}`

  When using before-login, you have to select the save-to-configuration option so that the change is saved to the configuration file to have it show up in output of the `show configuration {`*`module-name`*`} {`**`detail`**`}` command.

  When you use the **`acknowledge`** option, users must press a key to get the login prompt.

  When no options are specified, the command configures a banner for a CLI session that appears before login. A CLI banner can have a maximum size of 24 rows with 79 columns of text.
- To clear a configured banner, use the following command:
  `unconfigure banner { after-login | before-login }` command.
- To disable the acknowledgment feature (which forces the user to press a key before the login screen appears), use the following command omitting the **`acknowledge`** option:
  `configure banner {`**`after-login`** `| { `**`before-login`** `} { `**`acknowledge`** `} | `**`before-login`** `{`**`acknowledge`**`} `**`save-to-configuration`**`}`
- To display the banners that are configured on the switch, use the following command:
  `show banner { after-login | before-login }`

Startup Screen and Prompt Text

Once you log into the switch, the system displays the startup screen.

```
login: admin
password:

ExtremeXOS
Copyright (C) 1996-2023 Extreme Networks.
All rights reserved.
This product is protected by one or more US
patents listed at
http://www.extremenetworks.com/patents
along with their foreign counterparts.
================================================================================

Press the <tab> or '?' key at any time for completions.
Remember to save your configuration changes.

* <switchname>.1 #
```

You must have an administrator-level account to change the text of the prompt. The prompt text is taken from the *SNMP* sysname setting.

The number that follows the period after the switch name indicates the sequential line of the specific command or line for this CLI session.

If an asterisk (*) appears in front of the command line prompt, it indicates that you have outstanding configuration changes that have not been saved.

For example: `* 5420F-48P-4XE.1 #`

If you have logged on with administrator capabilities, the command line prompt ends with a (#) sign.

For example: `5420F-48P-4XE.1 #`

If you have logged on with user capabilities, the command line prompt ends with a (>) sign.

For example: `5420F-48P-4XE.1 >`

Default Accounts

By default, the switch is configured with two accounts. ExtremeXOS 15.7.1 added the ability to disable all default accounts ("admin" and "user").

Table 8: Default Accounts

Account Name	Access Level
admin	This user can access and change all manageable parameters. However, the user may not delete all admin accounts.
user	This user can view (but not change) all manageable parameters, with the following exceptions: • This user cannot view the user account database. • This user cannot view the *SNMP* community strings.

Creating Management Accounts

An account can be disabled or enabled locally using read/write access. Even all administrative privileged accounts and user privileged accounts can be disabled. Lawful-Intercept account will be disabled under user privileged option. When all administrative accounts will be disabled locally, a warning will be shown to use failsafe account, if necessary.

1. Log in to the switch as admin.
2. At the password prompt, press **[Enter]**, or enter the password that you have configured for the admin account.
3. Run the `create account [admin | user | ] account-name {encrypted encrypted_password | password}` command to add a new user.

   If you do not specify a password or the keyword encrypted, you are prompted for one. Passwords are case-sensitive.

   Caution
   Using the encrypted option incorrectly can result in being locked out of your switch account. For more information about the encrypted option, see create account.

   If you do not want a password associated with the specified account, press **[Enter]** twice.

   User-created account names are not case-sensitive.

Viewing Accounts

You can view all accounts. To view the accounts that have been created, you must have administrator privileges. Run the `show accounts` command.

Deleting an Account

You can remove accounts that should no longer exist, but you must have administrator privileges. To delete an account, run the `delete account` command.

Authenticating Management Sessions through the Local Database

You can use a local database on each switch to authenticate management sessions. An account can be disabled or enabled locally using read/write access. Even all administrative privileged accounts and user privileged accounts can be disabled. Lawful-Intercept account will be disabled under user privileged option. When all administrative accounts will be disabled locally, a warning will be shown to use failsafe account, if necessary.

This enable/disable command affects the following North Bound Interfaces (NBIs) in mgmt-access realm:

- console
- TELNET
- SSH
- HTTP
- XML

The local database stores user names and passwords and helps to ensure that any configuration changes to the switch can be done only by authorized users.

You can increase authentication security using Secure Shell 2 (SSH2). SSH2 provides encryption for management sessions. For information about SSH2, see Secure Shell 2 on page 1644.

Failsafe Accounts

The failsafe account is last possible method to access your switch.

This account is never displayed by the `show accounts` command, but it is always present on the switch. To display whether the user configured a username and password for the failsafe account, or to show the configured connection-type access restrictions, use the following command: `show failsafe-account .`

The failsafe account has admin access level.

To configure the account name and password for the failsafe account, use the following command:

`configure failsafe-account {[`**`deny`** `|` **`permit`**`] [`**`all`** `|` **`control`** `|` **`serial`** `|` **`ssh`** `{`**`vr`** *`vr-name`*`} |` **`telnet`** `{`**`vr`** *`vr-name`*`}]}`

When you use the command with no parameters, you are prompted for the failsafe account name and prompted twice to specify the password for the account.

For example:

```
# configure failsafe-account
enter failsafe user name: blue5green
enter failsafe password:
enter password again:
```

When you use the command with the permit or deny parameter, the connection-type access restrictions are altered as specified. For example:

```
# configure failsafe-account deny all
# configure failsafe-account permit serial
```

The failsafe account is immediately saved to NVRAM. On a SummitStack, the failsafe account is saved in the NVRAM of every node in the active topology.

Note
On a SummitStack, when the `synchronize stacking {node-address` *`node-address`* `| slot` *`slot-number`* `}` command is used, the failsafe account is transferred from the current node to the specified nodes in the stack topology.

You do not need to provide the existing failsafe account information to change it.

Note
The information that you use to configure the failsafe account cannot be recovered by Extreme Networks. Technical support cannot retrieve passwords or account names for this account. Protect this information carefully.

Accessing the Switch using Failsafe Account

You can access your switch using the failsafe account.

1. Connect to the switch using one of the (configured) permitted connection types.
2. At the switch login prompt, carefully enter the failsafe account name.

   If you enter an erroneous account name, you cannot re-enter the correct name. In that case, press **[Enter]** until you get a login prompt and then try again.
3. When prompted, enter the password.

Managing Passwords

When you first access the switch, you have a default account.

You configure a password for your default account. As you create other accounts (see Creating Management Accounts on page 54), you configure passwords for those accounts.

The software allows you to apply additional security to the passwords. You can enforce a specific format and minimum length for the password. Additionally, you can age out the password, prevent a user from employing a previously used password, and lock users out of the account after three consecutive failed login attempts.

You can change the password to an encrypted password after you create an account.

Applying a Password to the Default Account

Default accounts do not have passwords assigned to them. Passwords can have a minimum of zero and a maximum of 32 characters. (If you specify the format

of passwords using the `configure account password-policy char-validation` command, the minimum is eight characters.)

Note
Passwords are case-sensitive. User-created account names are not case-sensitive.

1. Log in to the switch using the name admin or user.
2. At the password prompt, press **[Enter]**.
3. Add a default admin password of green to the admin account or blue to the user account.

```
configure account admin green
configure account user blue
```

Note
If you forget your password while logged out of the CLI, you can use the bootloader to reinstall a default switch configuration, which allows access to the switch without a password. Note that this process reconfigures all switch settings back to the initial default configuration.

Applying Security to Passwords

You can increase the security of your system by enforcing password restrictions, which will make it more difficult for unauthorized users to access your system. You can specify that each password must include at least two characters of each of the following four character types:

- Upper-case A-Z
- Lower-case a-z
- 0-9
- !, @, #, $, %, ^, *, (,)

You can enforce a minimum length for the password, and set a maximum and minimum time limit, after which the password will not be accepted.

By default, lockout-on-login-failures is not enabled. The default settings are as follows:

- `configure account all password-policy lockout-on-login-failures off`
- `configure account all password-policy lockout-time-period until-cleared`

To increase security, you can lock users out of the system entirely after three failed consecutive logon attempts.

After the user's account is locked out (using the **`configure account password-policy lockout-on-login-failure`** command), it must be re-enabled by an administrator.

Version 33.1.1 adds additional restrictions for more secure user and password combinations. These include the following:

- A user name and password cannot be the same.
- The same letters or numbers cannot appear in succession in the pass phrase (no '11' or 'aa' in the passphrase).
- Sequential input (logical and keyboard indexed) beyond 3 characters is prohibited. For example, 1234, abcd, qwer, zxcv.
- Any password used within the last three months is prohibited.
- To set character requirements for the password, use the following command:

  `configure account [all | name] password-policy char-validation [none | all-char-groups]`
- To configure the number of characters in a revised password that must be changed from an existing password.

  `configure account [all |name] password-policy min-different-characters [count]`
- To set a minimum length for the password, use the following command:

  `configure account [all | name] password-policy min-length [num_characters | none]`
- To age out the password after a specified time, use the following command:

  configure account [all | name] password-policy max-age [num_days | none]
- To configure a minimum password lifespan, use the following command:

  `configure account [all | name] password-policy min-age [num_days | none]`
- To configure that the same letters or numbers can't appear in succession in the passphrase, use the following command:

  configure account [all | name] password-policy char-repeat [permit | deny]
- To configure that the user name and pass phrase cannot match, use the following command:

  `configure account [all | name] password-policy username-match [permit | deny]`
- To configure that sequential input (logical and keyboard indexed) beyond 3 characters is prohibited, use the following command:

  configure account [all | name] password-policy long-sequence [permit | deny]
- To block users from employing previously used passwords, use the following command:

  `configure account [all | name] password-policy history [num_passwords | duration days | none]`

- To disable an account after three consecutive failed logon attempts, use the following command:

  `configure account [all | name] password-policy lockout-on-login-failures [on | off]`

  Note
  If you are not working on SSH, you can configure the number of failed logons that trigger lockout, using the `configure cli max-failed-logins num-of-logins` command. (This command also sets the number of failed logons that terminate the particular session.)

- To re-enable a locked-out account, use the following command:

  `clear account [all | name] lockout`

  Selecting the all option affects the setting of all existing and future new accounts.
- To view account settings, use the following command:

  `show accounts password-policy`

Hash Algorithm for Account Passwords

SHA-256 is hash for local passwords. This hash is visible in both the XML config file (.cfg) and ASCII (.xsf). All existing users (created with older software) are still recognized and their *MD5 (Message-Digest algorithm 5)* hashes can be verified.

However, if a new user is created or a password is changed, it will use the SHA-256 hash. After a downgrade, older software will not be able to validate users with the SHA-256 hash. Upgrading will not automatically change the hash for existing users.

Removal of Cleartext Passwords

All passwords, secrets, or keys are not shown in the clear. The software does not emit those passwords in any commands or display them as part of the device configuration.

A new mode of operation for the CLI requires prompting (with no echo) for all passwords, secrets, or keys. The command `configure cli password prompting-only` controls this mode.

Each CLI command with password arguments is modified to use the new mode (designated with flags="prompting-only" in the CLI syntax attribute specification). Then prompting must be handled by that command.

Timed Lockout

This feature includes the option to disable an account for a configurable period of time after consecutive failed logins. After the configured duration elapses a disabled account is re-enabled automatically. The configurable period of lockout time ranges from 1 minute to 1 hour. The configurable number of the consecutive failed attempts ranges from 1 to 10.

The intent is to ensure the box is not ever completely locked out.

This feature augments this behavior in two ways:

- The failsafe account can now be locked out provided that the lockout is timed.
- All admin accounts can now be locked out provided that at least one is timed.

The feature applies to Telnet/SSH/Console/Https/Http.

Displaying Passwords

To display the accounts and any applied password security, use the following command:

- To display accounts and passwords, use the following command:

  ```
  show accounts password-policy
  ```

- To display which accounts can be locked out, use the following command:

  ```
  show accounts
  ```

Accessing an Active Node in a SummitStack

You can access any active node in a SummitStack from any other active node in the active topology by running:

```
telnet slot slot-number {no-auto-login}
```

You are not prompted for your username or password. You are logged in to the same account (with corresponding rights) with which you accessed the originating slot.

DNS Cache Resolver and Analytics Engine Overview

The Domain Name System (DNS) cache resolver feature implements a cache of DNS queries on the switch, so that repeated queries can be handled directly by the switch, rather than by repeatedly forwarding the requests to the DNS servers, which consumes time and network resources.

The DNS analytics engine analyzes the DNS queries (IPv4 and IPv6) from all connected clients and keeps track of received DNS queries from clients, and domains accessed along with time stamps. By using the cache and analytics, audits can be performed on the details of queries coming from clients, which allows for threat mitigation.

To provide GDPR compliance, DNS analytics provides a mechanism to restrict the collection of DNS analytics of a protected client. When you configure the client IP address subnet in the protected list, DNS queries from configured protected clients are erased from the analytics database and future queries are not stored.

Supported Platforms

All ExtremeSwitching Universal platforms.

Limitations

- TCP DNS queries are not cached.
- The DNS cache feature and the L7 DNS feature in ONEPolicy should not be enabled at the same time.
- Checkpointing is not supported in a stack or a MLAG setup for DNS caching.
- IPv6 DNS queries are not cached.
- DNS cache feature should not be enabled on VXLAN tenant VLANs.

Domain Name Service Client Services

The Domain Name Service (DNS) client augments the following commands to allow them to accept either IP addresses or host names.

- telnet
- download bootrom
- download image
- ping
- traceroute
- configure radius server client-ip
- configure tacacs server client-ip
- create cfm domain dns md-level

The DNS client can resolve host names to both IPv4 and IPv6 addresses. In addition, you can use the nslookup utility to return the IP address of a host name.

Use the following command to specify up to eight DNS servers for use by the DNS client:

```
configure dns-client add
```

Use the following command to specify a default domain for use when a host name is used without a domain.

```
configure dns-client default-domain
```

For example, if you specify the domain xyz-inc.com as the default domain, then a command such as `ping accounting1` is taken as if it had been entered `ping accounting1.xyz-inc.com`.

Checking Basic Connectivity

To check basic connectivity to your switch, use the `ping` and `traceroute` commands.

Ping

The ping command enables you to send *ICMP (Internet Control Message Protocol)* echo messages to a remote IP device.

The ping command is available for both the user and administrator privilege levels.

`ping {vr vr-name} {continuous|count|dont-fragment|interval|start-size|tos|ttl|udp} {mac|mpls|ipv4|ipv6} {from|with}`

Table 9: Ping Command Parameters

Parameter	Description
count	Specifies the number of ping requests to send.
start-size	Specifies the size, in bytes, of the packet to be sent, or the starting size if incremental packets are to be sent.
continuous	Specifies that UDP or ICMP echo messages are to be sent continuously. This option can be interrupted by pressing **[Ctrl]** + **C**.
end-size	Specifies an end size for packets to be sent.
udp	Specifies that the ping request should use UDP instead of ICMP.
dont-fragment	Sets the IP to not fragment the bit.
ttl	Sets the TTL value.
tos	Sets the TOS value.
interval	Sets the time interval between sending out ping requests.
vrid	Specifies the name to use for sending out the echo message. If not specified, *VR-Default* is used. **Note:** User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.
ipv4	Specifies IPv4 transport.
ipv6	Specifies IPv6 transport. **Note:** If you are contacting an IPv6 link local address, you must specify the *VLAN* you are sending the message from: `ping ipv6 link-local address %vlan_name host .`
host	Specifies a host name or IP address (either v4 or v6).
from	Uses the specified source address. If not specified, the address of the transmitting interface is used.
with record-route	Sets the traceroute information.

If a ping request fails, the switch stops sending the request after three attempts. Press **[Ctrl]** + **[C]** to interrupt a ping request earlier. The statistics are tabulated after the ping is interrupted or stops.

Use the ipv6 variable to ping an IPv6 host by generating an ICMPv6 echo request message and sending the message to the specified address. If you are contacting an IPv6 link local address, you must specify the VLAN that you are sending the message from, as shown in the following example (you must include the % sign):

```
ping ipv6 link-local address %vlan_name host
```

Traceroute

The `traceroute` command enables you to trace the path between the switch and a destination endstation.

```
traceroute {vr vrid} {ipv4 host} {ipv6 host} {ttl number} {from from}
{[port port] | icmp}
```

vr

The name of the .

ipv4/ipv6

The transport.

from

Uses the specified source address in the *ICMP* packet. If not specified, the address of the transmitting interface is used.

host

The host of the destination endstation. To use the hostname, you must first configure DNS.

ttl

Configures the switch to trace the hops until the time-to-live has been exceeded for the switch.

port

Uses the specified UDP port number.

icmp

Uses ICMP echo messages to trace the routed path.

Displaying Switch Information

You can display basic information about the switch by running the `show switch` command.

Filtering the Output of Show Commands

The output from many show commands can be long and complicated, sometimes containing more information than you need. You can filter the output from a show command to suit your needs using a restricted version of a UNIX/Linux feature that uses a "pipe" character to direct the output of one command to be used as input for the next command.

You can include or exclude show command output that matches your specified text pattern. You can also exclude all output until a line matches the pattern, and then include all output beginning with that line.

Most show commands can be filtered, but a few, for example, `show tech-support`, do not allow filtering.

To filter show command output, use the following command:

show *specific show command syntax* | [[**begin** | **exclude** | **grep** | **include**] { **ignore-case** } *regexp* { | **count** } | **count**]

show *specific show command syntax*	Specify the show command followed by the vertical bar (|) or pipe character. For example, `show ports`.
include	Display the lines that match the regular expression.
exclude	Do not display the lines that match the regular expression.
begin	Display all the lines starting with the first line that matches the regular expression.
grep	Display the lines that match the regular expression.
ignore-case	Ignore case distinctions in both the output and the regular expression.
count	Display the number of lines after the filtered output.
count	Display the number of lines after the output.
regexp	The text pattern (regular expression) to match (case-sensitive), enclosed by quotation marks.

If the specified show command outputs a refreshed display, filtering the show command terminates the display without refreshing and a message appears stating this.

Table 10 shows a summary of special characters.

Table 10: Definition of Regular Expression Characters

Operator Type	Examples	Description
Literal characters match a character exactly	a A y 6 % @	Letters, digits and many special characters match exactly
	\$ \^ \+ \\ \?	Precede other special characters with a \ to cancel their regex special meaning
	\n \t \r	Literal new line, tab, return
Anchors and assertions	^	Starts with
	$	Ends with
Character groups any one character from the group	[aAeEiou]	Any character listed from [to]
	[^aAeEiou]	Any character except aAeEio or u
	[a-fA-F0-9]	Any hex character (0 to 9 or a to f)
	.	Any character at all

Table 10: Definition of Regular Expression Characters (continued)

Operator Type	Examples	Description
Counts apply to previous element	+	One or more ("some")
	*	Zero or more ("perhaps some")
	?	Zero or one ("perhaps a")
Alternation	|	Either, or

Flow control

The following example display the status of “flow control” on the ports of a switch:

```
show ports 1-2 information detail | include "(Port|Flow Control)"
```

The output is similar to the following:

```
Port: 1
            Flow Control:   Rx-Pause: Enabled       Tx-Pause: Disabled
Priority Flow Control: Disabled

Port: 2
            Flow Control:   Rx-Pause: Enabled       Tx-Pause: Disabled
Priority Flow Control: Disabled
```

Automate Running Show Commands

This feature runs the integrated Python script `watch.py` to run a specified show command periodically and display the output accordingly. In the script, CLI paging is disabled so that user input is not needed, and CLI refreshing is disabled so that the keyword “no-refresh” is not needed in auto-refreshing commands.

To run a specified show command, use the following command:

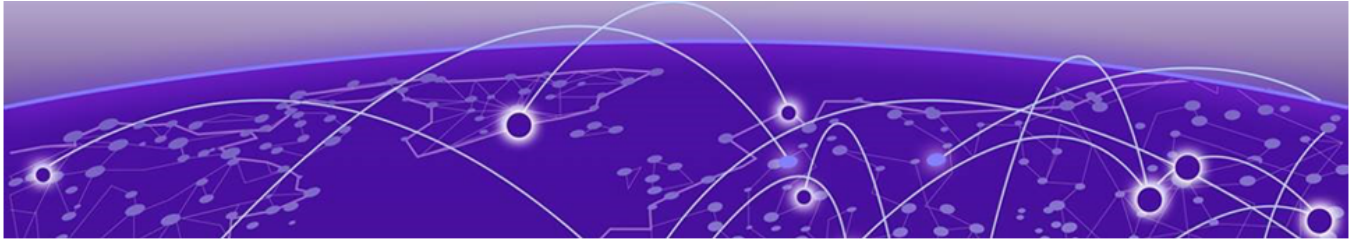
`watch` *command* {**difference**} {**count** *count*} {**interval** *interval*}.

Example

The following example runs the show ports command for port 11 runs 15 times with an interval of 30 seconds:

```
# watch "show ports 11 statistics port-number no-refresh" count 15 interval 30
Port      Link       Tx Pkt     Tx Byte      Rx Pkt     Rx Byte      Rx Pkt      Rx
Pkt      Tx Pkt      Tx Pkt

          State       Count       Count       Count       Count       Bcast
Mcast       Bcast       Mcast

========= ===== =========== =========== =========== =========== =========== ===========
=========== ===========

11        A          206845    30061110      205326    30939540           0
163381           0      163483

========= ===== =========== =========== =========== =========== =========== ===========
```

```
========== ==========

          > in Port indicates Port Display Name truncated past 8 characters

          > in Count indicates value exceeds column width. Use 'wide' option or '0' to
clear.

          Link State: A-Active, R-Ready, NP-Port Not Present, L-Loopback


Port      Link       Tx Pkt     Tx Byte      Rx Pkt     Rx Byte      Rx Pkt      Rx
Pkt      Tx Pkt      Tx Pkt

          State       Count       Count       Count       Count       Bcast
Mcast       Bcast       Mcast

========= ===== ========== ========== ========== ========== ========== ==========
========== ==========

11        A          206845    30061110      205327    30939662           0
163382           0      163483

========= ===== ========== ========== ========== ========== ========== ==========
========== ==========

          > in Port indicates Port Display Name truncated past 8 characters

          > in Count indicates value exceeds column width. Use 'wide' option or '0' to
clear.

          Link State: A-Active, R-Ready, NP-Port Not Present, L-Loopback


Port      Link       Tx Pkt     Tx Byte      Rx Pkt     Rx Byte      Rx Pkt      Rx
Pkt      Tx Pkt      Tx Pkt

          State       Count       Count       Count       Count       Bcast
Mcast       Bcast       Mcast

========= ===== ========== ========== ========== ========== ========== ==========
========== ==========

11        A          206847    30061402      205328    30939832           0
163383           0      163485

========= ===== ========== ========== ========== ========== ========== ==========
========== ==========

          > in Port indicates Port Display Name truncated past 8 characters

          > in Count indicates value exceeds column width. Use 'wide' option or '0' to
clear.

          Link State: A-Active, R-Ready, NP-Port Not Present, L-Loopback
```



Managing the Switch

ExtremeXOS Switch Management Overview on page 67
Understanding the Operating System Shell on page 68
Using the Console Interface on page 68
Using the 10/100 or 10/100/1000 Ethernet Management Port on page 69
Managing the Switch with Bluetooth Devices on page 70
Using ExtremeCloud IQ Site Engine to Manage the Network on page 71
Authenticating Users on page 71
Using Telnet on page 72
Using Secure Shell 2 on page 81
Using the Trivial File Transfer Protocol on page 83
Understanding System Redundancy on page 84
Understanding Hitless Failover Support on page 89
Understanding Power Supply Management on page 97
Using the Network Time Protocol on page 98
Using the Simple Network Management Protocol on page 106
Using the Simple Network Time Protocol on page 126
Access Profile Logging for HTTP/HTTPS on page 130
Enabling and Disabling Switch Management Access on page 132

This chapter provides information about how to use your ExtremeXOS switch. Included you will find information about the ExtremeXOS Shell, system redundancy, power supply management, user authentication, Telnet, and hitless failover support, as well as *SNMP (Simple Network Management Protocol)* and *SNTP (Simple Network Time Protocol)* usage information.

ExtremeXOS Switch Management Overview

This chapter describes how to use the operating system to manage the switch. It also provides details on how to perform the following various basic switch functions:

- Access the command line interface (CLI) by connecting a terminal (or workstation with terminal-emulation software) to the console port.
- Access the switch remotely using TCP/IP through one of the switch ports, or through the dedicated 10/100 unshielded twisted pair (UTP) Ethernet management port. Remote access includes:
  - Telnet using the CLI interface
  - Secure Shell (SSH2) using the CLI interface

  - *SNMP* access using an SNMP manager
- Download software updates and upgrades. For more information, see Software Upgrade and Boot Options on page 1697.

The switch supports the following number of concurrent user sessions:

- One console session
- Eight shell sessions
- Eight Telnet sessions
- Eight Trivial File Transfer Protocol (TFTP) sessions
- Eight SSH2 sessions
- Six XML sessions

Understanding the Operating System Shell

When you log in to the operating system from a terminal, a shell prompt is displayed.

At the prompt, input the commands you want to execute on the switch. After the switch processes and executes a command, the results are displayed on your terminal.

The shell supports ANSI, VT100, and XTERM terminal emulation and adjusts to the correct terminal type and window size. In addition, the shell supports UNIX-style page view for page-by-page command output capability.

By default, up to eight active shell sessions can access the switch concurrently; however, you can change the number of simultaneous, active shell sessions supported by the switch. You can configure up to 16 active shell sessions. Configurable shell sessions include both Telnet and SSH connections (not console CLI connections). If only eight active shell sessions can access the switch, a combination of eight Telnet and SSH connections can access the switch even though Telnet and SSH each support eight connections. For example, if you have six Telnet sessions and two SSH sessions, no one else can access the switch until a connection is terminated or you access the switch through the console.

If you configure a new limit, only new incoming shell sessions are affected. If you decrease the limit and the current number of sessions already exceeds the new maximum, the switch refuses only new incoming connections until the number of shell session drops below the new limit. Already connected shell sessions are not disconnected as a result of decreasing the limit.

Configure the number of shell sessions accepted by the switch, use the following command:

```
configure cli max-sessions
```

For more information about the line-editing keys that you can use with the ExtremXOS shell, see Line-Editing Keys on page 44.

Using the Console Interface

You can access the switch as needed through the command line interface.

The switch is accessible using the following connectors:

- Summit switches: 9-pin, RS-232 ports.

On a stand-alone switch, the console port is located on the front panel.

Note
For more information on the console port pinouts, see the *hardware installation guide* for your switch.

After the connection is established, you will see the switch prompt and can now log in.

Using the 10/100 or 10/100/1000 Ethernet Management Port

The switches provide a dedicated 10/100 Mbps or 10/100/1000 Mbps Ethernet management port. This port provides dedicated remote access to the switch using TCP/IP. It supports the following management methods:

- Telnet/SSH2 using the CLI interface
- *SNMP* access using an SNMP manager

The switch uses the Ethernet management port only for host operation, not for switching or routing. The TCP/IP configuration for the management port is done using the same syntax as used for *VLAN (Virtual LAN)* configuration. The VLAN management comes preconfigured with only the management port as a member. The management port is a member of the *VR-Mgmt*.

Most 5320 models do not have dedicated management ports. You can use front panel ports for management connectivity for these switches. Models 5320-24T-4X-XT and 5320- 24T-24S-4XE-XT have dedicated management ports.

On a SummitStack, the master node is accessed using the management port primary IP address for other platforms. The primary IP address is acquired by the backup node when it becomes the master node due to a failover. You can also directly access any node in the stack using its alternate IP address if the node's management port is connected to your network.

- To configure the IP address and subnet mask for the VLAN mgmt, use the following command:

  ```
  configure vlan mgmt ipaddress ip_address /subnet_mask
  ```

- To configure the default gateway (you must specify VR-Mgmt for the management port and VLAN mgmt), use the following command:

  ```
  configure iproute add default gateway { metric } {multicast |
  multicast-only | unicast | unicast-only} {vr vrname}
  ```

  The following example configuration sets the management port IP address to 192.168.1.50, mask length of 25, and configures the gateway to use 192.168.1.1:

  ```
  configure vlan mgmt ipaddress 192.168.1.50/25
  configure iproute add default 192.168.1.1 vr vr-mgmt
  ```

For more information see Logging into a Stack on page 183.

Managing the Switch with Bluetooth Devices

You can manage certain switches using Bluetooth with mobile devices or laptops. Bluetooth is enabled by default.

Supported Platforms

Switches: All ExtremeSwitching Universal platforms.

Adapters: Dongles with a common Bluetooth controller (Cypress CY20702). Vendors with CY20702 pluggable and CC&C (USB-BT-400).

Limitations

- Stacking with Bluetooth is not supported. Currently, access to a slot connected with a Bluetooth device is supported, but other slots cannot be accessed.
- No security procedures are initiated for Bluetooth connectivity.
- Only a specific list of Bluetooth dongles is supported.
- Only one Bluetooth adapter is enabled per card. Other adapters will be in powered off state.
- On platforms that do not support VR-Mgmt (for example, ExtremeSwitching 5320 series switches), the DHCP server will not run on the Bluetooth interface. Instead, clients can connect using the Link Local IPv4 address only. Clients that do not support Link Local IPv4 address (such as Mobile) are not able to access the switch.

Using Bluetooth

Bluetooth users/clients can access switches using either the global IP address or link local IP address. The Bluetooth interface is assigned with the global IP address (192.168.1.1/24 or 172.16.1.1/24). DHCP server runs on the Bluetooth interface. This is mainly to serve Bluetooth clients with DHCP support. After a Bluetooth connection is established, all Bluetooth clients are allocated IP addresses from this pool (192.168.1.2–192.168.1.20 or 172.16.1.1–172.16.1.20). For Bluetooth clients that do not support DHCP, you can use the link local IP address for accessing the switch. Range is 169.254.1.1–169.254.1.8.

To set up a Bluetooth device to manage a switch:

1. Insert a supported dongle into the USB port of the switch.
2. If needed (Bluetooth is enabled by default), ensure that Bluetooth capability, discovery, and pairing are all enabled. using the following command:

   ```
   enable switch bluetooth {discovery | pairing }
   ```

   You can check Bluetooth status using the `show switch bluetooth [statistics | inventory]` command.
3. If needed, enable Bluetooth on the desired Bluetooth device (phone or laptop).
4. To initiate pairing, do one of the following:
   - Press and hold the **mode** button for approximately 5 seconds.

- Press the pairing button on the dongle.

The Bluetooth Status LED blinks green while pairing is in progress, and then becomes solid green when Bluetooth is connected.

Bluetooth Commands

To enable Bluetooth capabilities, use the following command:

```
enable switch bluetooth {discovery | pairing }
```

To disable Bluetooth capabilities, use the following command:

```
disable switch bluetooth {discovery | pairing }
```

```
clear switch bluetooth device [all | address]
```

To show Bluetooth information, use the following command:

```
show switch bluetooth [statistics | inventory]
```

Using ExtremeCloud IQ Site Engine to Manage the Network

ExtremeCloud IQ Site Engine is a powerful yet easy-to-use application suite that facilitates the management of a network of Extreme Networks switches, as well as selected third-party switches.

This product offers a comprehensive set of network management tools that are easy to use from a client workstation running client software, or from a workstation configured with a web browser and the Java plug-in.

Authenticating Users

There are three methods to authenticate users who log in to the switch: *RADIUS (Remote Authentication Dial In User Service)* client, TACACS+, and a local database of accounts and passwords.

Note

You cannot configure RADIUS and TACACS+ at the same time.

RADIUS Client

Remote Authentication Dial In User Service (*RADIUS*, RFC 2865) is a mechanism for authenticating and centrally administrating access to network nodes.

The ExtremeXOS RADIUS client implementation allows authentication for Telnet or console access to the switch. For detailed information about RADIUS and configuring a RADIUS client, see Security on page 1628.

TACACS+

Terminal Access Controller Access Control System, often run on UNIX systems,provides access control for routers, network access servers, and other networked computing devices via one or more centralized servers. TACACS+ provides separate authentication, authorization, and accounting services. User passwords are administered in a central database rather than in individual routers, providing easily scalable network security solutions.

Management Accounts

ExtremeXOS supports two levels of management accounts (local database of accounts and passwords): user and administrator.

A user level account can view but not change all manageable parameters, with the exception of the user account database and *SNMP* community strings. An administrator level account can view and change all manageable parameters.

For detailed information about configuring management accounts, see Configuring Management Access on page 50.

Using Telnet

The operating system supports the Telnet Protocol based on RFC 854.

Telnet allows interactive remote access to a device and is based on a client/server model. The operating system uses Telnet to connect to other devices from the switch (client) and to allow incoming connections for switch management using the CLI (server).

Starting the Telnet Client

Ensure that the IP parameters described in Configuring Switch IP Parameters on page 73 are set up and then start an outgoing Telnet session.

Telnet is enabled and uses *VR-Mgmt* by default.

Note

Maximize the Telnet screen so that it correctly displays screens that automatically update.

1. Use Telnet to establish a connection to the switch.
2. Specify the IP address or host name of the device that you want to connect to.

   Check the user manual supplied with the Telnet facility if you are unsure of how to do this.

   After the connection is established, you see the switch prompt and you can log in. The same is true if you use the switch to connect to another host. From the CLI, you must specify the IP address or host name of the device that you want to connect to.
3. If the host is accessible and you are allowed access, you may log in.

For more information about using the Telnet client on the switch, see Connect to Another Host Using Telnet on page 73.

About the Telnet Server

Any workstation with a Telnet facility should be able to communicate with the switch over a TCP/IP network using VT100 terminal emulation.

Up to eight active Telnet sessions can access the switch concurrently. If you enable the idle timer using the `enable cli idle-timeout` command, the Telnet connection times out after 20 minutes of inactivity by default. If a connection to a Telnet session is lost inadvertently, the switch terminates the session within two hours.

The switch accepts IPv6 connections.

For information about the Telnet server on the switch, see the following sections:

- Configuring Telnet Access to the Switch on page 76
- Disconnecting a Telnet Session on page 77

Connect to Another Host Using Telnet

You can Telnet from the current CLI session to another host. For more information see Starting the Telnet Client on page 72.

Run `telnet {vr vr_name} [host_name | remote_ip] {port} .`

User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

If the TCP port number is not specified, the Telnet session defaults to port 23. If the name is not specified, the Telnet session defaults to *VR-Mgmt*. Only VT100 emulation is supported.

Configuring Switch IP Parameters

To manage the switch by way of a Telnet connection or by using an *SNMP* Network Manager, you must first configure the switch IP parameters.

Using a BOOTP or DHCP Server

The switch contains a BOOTP and *DHCP (Dynamic Host Configuration Protocol)* client, so if you have a BOOTP or DHCP server in your IP network, you can have it assign IP addresses to the switch. This is more likely to be desirable on the switch's *VLAN* mgmt than it is on any other VLANs.

If you are using IP and you have a Bootstrap Protocol (BOOTP) server set up correctly on your network, you must provide the following information to the BOOTP server:

- Switch Media Access Control (MAC) address, found on the rear label of the switch
- IP address

- Subnet address mask (optional)

Note

Dual network operating system ExtremeSwitching switches (for example, 5520 switches) when running ExtremeXOS use a base MAC address at offset 0 (for example: 00:c0:cc:8b:68:00) for both the OOB management port as well as inband VLANs using DHCP. However, when running VOSS, the offset is 0x81 for OOB (for example: 00:c0:cc:8b:68:81) and 256 (for example: 00:c0:cc:8b:69:00), assuming the first VLAN MAC offset 0 is assigned to the inband VLAN. If a different VLAN MAC offset is assigned, this MAC address changes accordingly (for example, if VLAN MAC offset is 10, then the associated MAC address is 00:c0:cc:8b:69:0A).

When using DHCP client on the switch, the switch sends a common DHCP client identifier equal to the base MAC address of the switch that is printed on the switch label. When transitioning between VOSS and ExtremeXOS, since the DHCP client ID is the same, the same IP address should be given out by the DHCP server assuming a standard DHCP pool configuration. If you want to statically assign IP addresses on the DHCP server, then it is recommended to assign them based on the DHCP client ID, since this ensures the binding does not change when alternating between VOSS and ExtremeXOS. If the DHCP IP addresses are assigned based on MAC addresses, multiple entries have to be configured (one for 0x81 offset, and one for :00 offset) to account for differences in the OOB or inband VLAN source MAC address usage between the two operating systems.

The switch does not retain IP addresses assigned by BOOTP or DHCP through a power cycle, even if the configuration has been saved. To retain the IP address through a power cycle, you must configure the IP address of the VLAN using the CLI or Telnet.

If you need the switch's MAC address to configure your BOOTP or DHCP server, you can find it on the rear label of the switch. Note that all VLANs configured to use BOOTP or DHCP use the same MAC address to get their IP address, so you cannot configure the BOOTP or DHCP server to assign multiple specific IP addresses to a switch depending solely on the MAC address.

- To enable the BOOTP or DHCP client per VLAN, use the following command:

  ```
  enable bootp vlan [ vlan_name | all]
  enable dhcp vlan [ vlan_name | all]
  ```

- To disable the BOOTP or DHCP client per VLAN, use the following command:

  ```
  disable bootp vlan [ vlan_name | all]
  disable dhcp vlan [ vlan_name | all]
  ```

- To view the current state of the BOOTP or DHCP client, use the following command:

  ```
  show dhcp-client state
  ```

Note

The ExtremeXOS DHCP client will discard the DHCP OFFER if the lease time is less than or equal to two seconds.

Manually Configuring the IP Settings

If you are using IP without a BOOTP server, you must enter the IP parameters for the switch in order for the *SNMP* Network Manager or Telnet software to communicate with the device.

1. Assign IP parameters to the switch.
   a. Log in to the switch with administrator privileges using the console interface.
   b. Assign an IP address and subnet mask to a *VLAN*.
   c. The switch comes configured with a default VLAN named default. To use Telnet or an SNMP Network Manager, you must have at least one VLAN on the switch, and that VLAN must be assigned an IP address and subnet mask. IP addresses are always assigned to each VLAN. The switch can be assigned multiple IP addresses (one for each VLAN).

      Note
      For information on creating and configuring VLANs, see VLANs on page 2213.

2. Manually configure the IP settings.
   a. Connect a terminal or workstation running terminal emulation software to the console port, as detailed in Using the Console Interface on page 68.
   b. At your terminal, press **[Enter]** one or more times until you see the login prompt.
   c. At the login prompt, enter your user name and password. The user name is not case-sensitive; the password is case-sensitive. Ensure that you have entered a user name and password with administrator privileges.

      If you are logging in for the first time, use the default user name *admin* to log in with administrator privileges. For example: `login: admin`

      Administrator capabilities enable you to access all switch functions. The default user names have no passwords assigned.

      If you have been assigned a user name and password with administrator privileges, enter them at the login prompt.

d. Enter the password when prompted.
   When you have successfully logged in to the switch, the command line prompt displays the name of the switch.
e. Assign an IP address and subnetwork mask for the default VLAN by using the following command:
   `configure {vlan} vlan_name ipaddress [ipaddress {ipNetmask} | ipv6-link-local | {eui64} ipv6_address_mask]`
   For example:
   ```
   configure vlan default ipaddress 123.45.67.8 255.255.255.0
   ```
   The changes take effect immediately.

Note
As a general rule, when configuring any IP addresses for the switch, you can express a subnet mask by using dotted decimal notation or by using classless inter domain routing notation (CIDR). CIDR uses a forward slash plus the number of bits in the subnet mask. Using CIDR notation, the command identical to the previous example is: `configure vlan default ipaddress 123.45.67.8/24`.

3. Configure the default route for the switch using the following command:
   `configure iproute add default gateway {metric} {multicast | multicast-only | unicast | unicast-only} {vr vrname}`
   For example:
   ```
   configure iproute add default 123.45.67.1
   ```
4. Save your configuration changes so that they will be in effect after the next switch reboot.
   If you want to save your changes to the currently booted configuration, use the following command: `save`

   ExtremeXOS allows you to select or create a configuration file name of your choice to save the configuration to.
   a. If you want to save your changes to an existing or new configuration file, use the following command:
      `save configuration {primary | secondary | existing-config | new-config}`
5. When you are finished using the facility, log out of the switch by typing: `logout` or `quit`.

Configuring Telnet Access to the Switch

By default, Telnet services are enabled on the switch and all virtual routers listen for incoming Telnet requests. The switch accepts IPv6 connections.

User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

The safe defaults mode runs an interactive script that allows you to enable or disable *SNMP*, Telnet, and switch ports. When you set up your switch for the first time, you

must connect to the console port to access the switch. After logging in to the switch, you will enter into the safe defaults mode. Although SNMP, Telnet, and switch ports are enabled by default, the script prompts you to confirm those settings.

If you choose to keep the default setting for Telnet—the default setting is enabled—the switch returns the following interactive script:

```
Since you have chosen less secure management methods, please remember to increase the
security of your network by taking the following actions:
* change your admin password
* change your SNMP public and private strings
* consider using SNMPv3 to secure network management traffic
```

For more detailed information about safe defaults mode, see Using Safe Defaults Mode on page 48.

- To configure the from which you receive a Telnet request, use the following command:

  `configure telnet vr [`**`all`** `|` **`default`** `|` *`vr_name`*`]`

- To change the default, use the following command:

  `configure telnet port [`*`portno`* `|` **`default`**`]`

  The range for the port number is 1–65535. The following TCP port numbers are reserved and cannot be used for Telnet connections: 22, 80, and 1023. If you attempt to configure a reserved port, the switch displays an error message.

Viewing Telnet Information

To display the status of Telnet, including the current TCP port, the used to establish a Telnet session, and whether ACLs are controlling Telnet access, run the `show switch management` command.

Disabling and Enabling Telnet

Note
You must be logged in as an administrator to configure the virtual router(s) used by Telnet and to enable or disable Telnet.

- You can choose to disable Telnet by using the following command:

  `disable telnet`

- To re-enable Telnet on the switch, use the following command:

  `enable telnet`

Disconnecting a Telnet Session

A person with an administrator level account can disconnect a Telnet management session.

1. Log in to the switch with administrator privileges.
2. Determine the session number of the session you want to terminate.

   `show session {{`**`detail`**`} {`*`sessID`*`}} {`**`history`**`}`

3. Terminate the session.

   ```
   clear session [history |sessId | all]
   ```

The user logged in by way of the Telnet connection is notified that the session has been terminated.

Access Profile Logging for Telnet

By default, Telnet services are enabled on the switch.

The access profile logging feature allows you to use an *ACL (Access Control List)* policy file or dynamic ACL rules to control access to Telnet services on the switch. When access profile logging is enabled for Telnet, the switch logs messages and increments counters when packets are denied access to Telnet. No messages are logged for permitted access.

You can manage Telnet access using one (not both) of the following methods:

- Create and apply an ACL policy file.
- Define and apply individual ACL rules.

One advantage of ACL policy files is that you can copy the file and use it on other switches. One advantage to applying individual ACL rules is that you can enter the rules at the CLI command prompt, which can be easier than opening, editing, and saving a policy file.

ACL Match Conditions and Actions

The ACLs section describes how to create *ACL* policies and rules using match conditions and actions. Access profile logging supports the following match conditions and actions:

- Match conditions
  - Source-address—IPv4 and IPv6
- Actions
  - Permit
  - Deny

If the ACL is created with more match conditions or actions, only those listed above are used for validating the packets. All other conditions and actions are ignored.

The source-address field allows you to identify an IPv4 address, IPv6 address, or subnet mask for which access is either permitted or denied.

If the Telnet traffic does not match any of the rules, the default behavior is deny.

Limitations

Access profile logging for Telnet has the following limitations:

- Either policy files or ACL rules can be associated with Telnet, but not both at the same time.
- Only source-address match is supported.

- Access-lists that are associated with one or more applications cannot be directly deleted. They must be unconfigured from the application first and then deleted from the CLI.
- Default counter support is added only for ACL rules and not for policy files.

Managing ACL Policies for Telnet

The ACLs section describes how to create *ACL* policy files.

1. To configure Telnet to use an ACL policy, use the following command:

   ```
   configure telnet access-profile profile_name
   ```

2. To configure Telnet to remove a previously configured ACL policy, use the following command:

   ```
   configure telnet access-profile none
   ```

Note
Do not also apply the policy to the access list. Applying a policy to both an access profile and an access list is neither necessary nor recommended.

Managing ACL Rules for Telnet

Before you can assign an *ACL* rule to Telnet, you must create a dynamic ACL rule as described in ACLs.

1. To add or delete a rule for Telnet access, use the following command:

   ```
   configure telnet access-profile [ access_profile | [[add rule ] [first
   | [[before | after] previous_rule]]] | delete rule | none ]
   ```

2. To display the access-list permit and deny statistics for an application, use the following command:

   ```
   show access-list counters process [snmp | telnet | ssh2 | http]
   ```

Misconfiguration Error Messages

The following messages can appear during configuration of policies or rules for the SNMP service:

Rule <rule> is already applied	A rule with the same name is already applied to this service.
Please remove the policy <policy> already configured, and then add rule <rule>	A policy file is already associated with the service. You must remove the policy before you can add a rule.
Rule <previous_rule> is not already applied	The specified rule has not been applied to the service, so you cannot add a rule in relation to that rule.
Rule <rule> is not applied	The specified rule has not been applied to the service, so you cannot remove the rule from the service.
Error: Please remove previously configured rule(s) before configuring policy <policy>	A policy or one or more ACL rules are configured for the service. You must delete the remove the policy or rules from the service before you can add a policy.

Sample ACL Policies

The following are sample policies that you can apply to restrict Telnet access.

In the following example named MyAccessProfile.pol, the switch permits connections from the subnet 10.203.133.0 /24 and denies connections from all other addresses:

```
MyAccessProfile.pol
entry  AllowTheseSubnets {
    if  {
        source-address 10.203.133.0 /24;
    } then {
        permit;
    }
}
```

In the following example named MyAccessProfile.pol, the switch permits connections from the subnets 10.203.133.0 /24 or 10.203.135.0/24 and denies connections from all other addresses:

```
MyAccessProfile.pol
entry  AllowTheseSubnets {
    if match any {
        source-address 10.203.133.0 /24;
        source-address 10.203.135.0 /24;
    } then {
        permit;
    }
}
```

In the following example named MyAccessProfile_2.pol, the switch does not permit connections from the subnet 10.203.133.0 /24 but accepts connections from all other addresses:

```
MyAccessProfile_2.pol
entry dontAllowTheseSubnets {
    if {
        source-address 10.203.133.0 /24;
    } then {
        deny;
    }
}
entry  AllowTheRest {
    if  {
        ; #none specified
    } then {
        permit;
    }
}
```

In the following example named MyAccessProfile_2.pol, the switch does not permit connections from the subnets 10.203.133.0/24 or 10.203.135.0 /24 but accepts connections from all other addresses:

```
MyAccessProfile_2.pol
entry dontAllowTheseSubnets {
    if match any {
        source-address 10.203.133.0 /24;
        source-address 10.203.135.0 /24;
    } then {
        deny;
    }
```

```
}
entry  AllowTheRest {
    if  {
        ; #none specified
    } then {
        permit;
    }
}
```

Using Secure Shell 2

Secure Shell 2 (SSH2) is a feature of the software that allows you to encrypt session data between a network administrator using SSH2 client software and the switch or send encrypted data from the switch to an SSH2 client on a remote system. Configuration, image, public key, and policy files can be transferred to the switch using the Secure Copy Protocol 2 (SCP2).

The SSH2 switch application works with the following clients: Putty, SSH2 (version 2.x or later) from SSH Communication Security, and OpenSSH (version 2.5 or later).

The switch accepts IPv6 connections.

Up to eight active SSH2 sessions can run on the switch concurrently. If you enable the idle timer using the `enable cli idle-timeout` command, the SSH2 connection times out after 20 minutes of inactivity by default. If you disable the idle timer using the `disable cli idle-timeout` command, the SSH2 connection times out after 60 minutes of inactivity, by default. You can modify the timeout value using the command `configure ssh2 idletimeout <minutes>` where <minutes> can be from 1 to 240 . For more information, refer to the `help` command for `configure ssh2 idletimeout`. If a connection to an SSH2 session is lost inadvertently, the switch terminates the session within 60 minutes.

For detailed information about SSH2, see Security on page 1628.

Access Profile Logging for SSH2

The access profile logging feature allows you to use an *ACL* policy file or dynamic ACL rules to control access to SSH2 services on the switch.

When access profile logging is enabled for SSH2, the switch logs messages and increments counters when packets are denied access to SSH2. No messages are logged for permitted access.

You can manage SSH2 access using one (not both) of the following methods:

- Create and apply an ACL policy file
- Define and apply individual ACL rules

One advantage of ACL policy files is that you can copy the file and use it on other switches. One advantage to applying individual ACL rules is that you can enter the rules at the CLI command prompt, which can be easier than opening, editing, and saving a policy file.

ACL Match Conditions and Actions

The ACLs section describes how to create *ACL* policies and rules using match conditions and actions. Access profile logging supports the following match conditions and actions:

- Match conditions
  - Source-address—IPv4 and IPv6
- Actions
  - Permit
  - Deny

If the ACL is created with more match conditions or actions, only those listed above are used for validating the packets. All other conditions and actions are ignored.

The source-address field allows you to identify an IPv4 address, IPv6 address, or subnet mask for which access is either permitted or denied.

If the Telnet traffic does not match any of the rules, the default behavior is deny.

Limitations

Access profile logging for SSH2 has the following limitations:

- Either policy files or ACLs can be associated with SSH2, but not both at the same time.
- Only source-address match is supported.
- Access-lists that are associated with one or more applications cannot be directly deleted. They must be unconfigured from the application first and then deleted from the CLI.
- Default counter support is added only for dynamic ACL rules and not for policy files.

Managing ACL Policies for SSH2

The ACLs section describes how to create *ACL* policy files.

- To configure SSH2 to use an ACL policy, use the following command:

  ```
  configure ssh2 access-profile profile_name
  ```

- To configure SSH2 to remove a previously configured ACL policy, use the following command:

  ```
  configure ssh2 access-profile none
  ```

Managing ACL Rules for SSH2

Before you can assign an *ACL* rule to SSH, you must create a dynamic ACL rule as described in ACLs.

- To add or delete a rule for SSH2 access, use the following command:

  ```
  configure ssh2 access-profile [ access_profile | [[add rule ] [first | [[before | after] previous_rule]]] | delete rule | none ]
  ```

- To display the access-list permit and deny statistics for an application, use the following command:

  ```
  show access-list counters process [snmp | telnet | ssh2 | http]
  ```

Misconfiguration Error Messages

The following messages can appear during configuration of policies or rules for the SNMP service:

Rule <rule> is already applied	A rule with the same name is already applied to this service.
Please remove the policy <policy> already configured, and then add rule <rule>	A policy file is already associated with the service. You must remove the policy before you can add a rule.
Rule <previous_rule> is not already applied	The specified rule has not been applied to the service, so you cannot add a rule in relation to that rule.
Rule <rule> is not applied	The specified rule has not been applied to the service, so you cannot remove the rule from the service.
Error: Please remove previously configured rule(s) before configuring policy <policy>	A policy or one or more ACL rules are configured for the service. You must delete the remove the policy or rules from the service before you can add a policy.

Using the Trivial File Transfer Protocol

ExtremeXOS supports the Trivial File Transfer Protocol (TFTP) based on RFC 1350.

TFTP is a method used to transfer files from one network device to another. The ExtremeXOS TFTP client is a command line application used to contact an external TFTP server on the network. For example, ExtremeXOS uses TFTP to download software image files, switch configuration files, and ACLs from a server on the network to the switch.

Up to eight active TFTP sessions can run on the switch concurrently.

We recommend using a TFTP server that supports blocksize negotiation (as described in RFC 2348, TFTP Blocksize Option), to enable faster file downloads and larger file downloads.

Note
For better functionality, minimum block-size of 64 bytes is recommended.

For additional information about TFTP, see the following chapters:

- For information about downloading software image files, BootROM files, and switch configurations, see Software Upgrade and Boot Options on page 1697.
- For information about downloading *ACL* (and other) policy files, see Policy Manager.
- For information about using TFTP to transfer files to and from the switch, see Managing the Switch on page 67.
- For information about configuring core dump files and transferring the core dump files stored on your switch, see Troubleshooting on page 2107

TFTP Block-size Configuration

ExtremeXOS supports the TFTP client block-size option configuration based on RFC 2348, which can range from 8 octets to 65000 octets. The block-size refers to data

octets and does not include TFTP header. This feature added the user configuration option for data bock-size:

- in the generic commands that are used for downloading/uploading image/ configuration/log/core file etc.
- ranging from 24 bytes to 65000 bytes, taking into consideration the local/remote file name size and the current busy box TFTP client support limits.
- to support larger TFTP data packets exceeding normal MTU especially on a front-panel port in case of in-band management, enable jumbo frames on that port. Please refer "Jumbo Frames" for the usage and its functional restrictions that affect TFTP data packet transfers.

If you do not specify block-size, the default is 8192 bytes.

Connecting to Another Host Using TFTP

You can TFTP from the current CLI session to another host to transfer files.

1. Run the `tftp` command:

   ```
   tftp [ ip-address | host-name ] { -v vr_name } { -b block_size } [ -g | -p ] [ -l local-file { -r remote-file } | -r remote-file { -l localfile } ]
   ```

   Note
   User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

   The TFTP session defaults to port 69. If you do not specify a , *VR-Mgmt* is used.

   For example, to connect to a remote TFTP server with an IP address of 10.123.45.67 and “get” or retrieve an ExtremeXOS configuration file named `XOS1.cfg` from that host, use the following command:

   ```
   tftp 10.123.45.67 -g -r XOS1.cfg
   ```

   When you “get” the file through TFTP, the switch saves the file.
2. To view the files you retrieved, run the `ls` command at the command prompt.

   In addition to the `tftp` command, the following two commands are available for transferring files to and from the switch:

   ```
   tftp get [ ip-address | host-name] { vr vr_name } { block-size block_size } remote-file local-file} {force-overwrite}
   ```

   By default, if you transfer a file with a name that already exists on the system, the switch prompts you to overwrite the existing file. For more information, see the `tftp get` command.

Understanding System Redundancy

If you configure two master-capable nodes in a SummitStack, one assumes the role of primary (also called "master") and the other assumes the role of backup.

The primary node provides all of the switch management functions including running the bridging and routing protocols, and configuring the switch. The primary node also synchronizes the backup node in case it needs to take over the management functions if the primary node fails.

For SummitStack, a node can be a redundant primary node if it has been configured to be master-capable.

To configure master capability on one or all nodes in a SummitStack, use one of the following commands:

```
configure stacking [node-address node-address | slot slot-number]
alternate-ip-address [ipaddress netmask | ipNetmask] gateway
configure stacking redundancy [none | minimal | maximal]
```

Node Election

Node election is based on leader election between master-capable nodes present in a SummitStack.

By default, the SummitStack node in slot 1 has primary status. Each node uses health information about itself together with a user configured priority value to compute its node role election priority. Nodes exchange their node role election priorities. During the node election process, the node with the highest node role election priority becomes the master or primary node, and the node with the second highest node role election priority becomes the backup node. All other nodes (if any) remain in STANDBY state.

The primary node runs the switch management functions, and the backup node is fully prepared to become the primary node if the primary fails. In SummitStack, nodes that remain in STANDBY state (called Standby nodes) program their port hardware based on instructions received from the primary. Standby nodes configured to be master-capable elect a new backup node from among themselves after a failover has occurred.

Determining the Primary Node

The following parameters determine the primary node:

Node state

The node state must be STANDBY to participate in leader election and be selected as primary. If the node is in the INIT, DOWN, or FAIL states, it cannot participate in leader election. For more information about the node states, see Viewing Node Status on page 89.

Configuration priority

This is a user assigned priority. The configured priority is compared only after the node meets the minimum thresholds in each category for it to be healthy. Required processes and devices must not fail.

Software health

This represents the percent of processes available.

Health of secondary hardware components

This represents the health of the switch components, such as power supplies, fans, and so forth.

Slot ID

The slot number configured on a stack node.

Configuring the Node Priority on a SummitStack

If you do not configure any priorities, slot 1 has the highest priority, slot 2 the second highest priority, and so forth in order of increasing slot number. You may also use the factory assigned MAC address as the node-address value. By default the priority is "automatic" and the node-pri value is any number between 1 and 100. The higher the value, the higher the priority.

Configure the priority of a node in a SummitStack using the following command:

```
configure stacking {node-address node-address | slot slot-number}
priority [node-pri | automatic]
```

Relinquishing Primary Status

Before relinquishing primary status and initiating failover, review the section Synchronizing Nodes on page 1721 to confirm that your platform and both installed MSMs/MMs or master-capable nodes are running software that supports the `synchronize` command.
You can cause the primary to failover to the backup, thereby relinquishing its primary status.

1. Use the `show switch {detail}` command on the primary or the backup node to confirm that the nodes are synchronized and have identical software and switch configurations before failover.

   A node may not be synchronized because checkpointing did not occur, incompatible software is running on the primary and backup, or the backup is down.

   If the nodes are not synchronized and both nodes are running a version of ExtremeXOS that supports synchronization, proceed to step 2.

   If the nodes are synchronized, proceed to step 3 on page 87.

   The output displays the status of the nodes, with the primary node showing MASTER and the backup node showing BACKUP (InSync).
2. If the nodes are not synchronized because of incompatible software, use the `synchronize` command to ensure that the backup has the same software in flash as the primary.

   The `synchronize` command:

   - Reboots the backup node to prepare it for synchronizing with the primary node.
   - Copies both the primary and secondary software images.
   - Copies both the primary and secondary configurations.
   - Reboots the backup node after replication is complete.

   After you confirm the nodes are synchronized, proceed to step 3.

3. If the nodes are synchronized, use the `run failover {force}` command to initiate failover from the primary node to the backup node.

   The backup node then becomes the primary node and the original primary node reboots.

Replicating Data Between Nodes

ExtremeXOS replicates configuration and run-time information between the primary node and the backup node so that the system can recover if the primary fails. This method of replicating data is known as checkpointing. Checkpointing is the process of automatically copying the active state from the primary to the backup, which allows for state recovery if the primary fails.

Replicating data consists of the following three steps:

- Configuration synchronization—Relays current and saved configuration information from the primary to the backup.
- Bulk checkpoint—Ensures that each individual application running on the system is synchronized with the backup.
- Dynamic checkpoint—Checkpoints any new state changes from the primary to the backup.

To monitor the checkpointing status, use the following command:

```
show checkpoint-data {process}
```

Data is not replicated from the primary to the standby nodes.

Relaying Configuration Information

To facilitate a failover from the primary node to the backup node, the primary transfers its active configuration to the backup.

Relaying configuration information is the first level of checkpointing . During the initial switch boot-up, the primary's configuration takes effect. During the initialization of a node, its configuration is read from the local flash. After the primary and backup nodes have been elected, the primary transfers its current active configuration to the backup. After the primary and backup nodes are synchronized, any configuration change you make to the primary is relayed to the backup and incorporated into the backup's configuration copy.

Note
To ensure that all of the configuration commands in the backup's flash are updated, issue the **save** command after you make any changes. On a SummitStack, the save configuration command will normally save the primary node's configuration file to all active nodes in the SummitStack.

If a failover occurs, the backup node continues to use the primary's active configuration. If the backup determines that it does not have the primary's active configuration because a run-time synchronization did not happen, the switch or SummitStack

reboots. Because the backup always uses the primary's active configuration, the active configuration remains in effect regardless of the number of failovers.

Note

If you issue the `reboot` command before you save your configuration changes, the switch prompts you to save your changes. To keep your configuration changes, save them before you reboot the switch.

Bulk Checkpointing

Bulk checkpointing causes the master and backup run-time states to be synchronized. Since ExtremeXOS runs a series of applications, an application starts checkpointing only after all of the applications it depends on have transferred their run-time states to the backup node.

After one application completes bulk checkpointing, the next application proceeds with its bulk checkpointing.

- To monitor the checkpointing status, use the `show checkpoint-data {process}` command.
- To see if bulk checkpointing is complete (that is, to see if the backup node is fully synchronized [in sync] with the master node), use the `show switch {detail}` command.

If a failover occurs before bulk checkpointing is complete, the SummitStack reboots. However, once bulk checkpointing is complete, failover is possible without a SummitStack reboot.

Dynamic Checkpointing

After an application transfers its saved state to the backup node, dynamic checkpointing requires that any new configuration information or state changes that occur on the primary be immediately relayed to the backup.

This ensures that the backup has the most up-to-date and accurate information.

Viewing Checkpoint Statistics

View and check the status of one or more processes being copied from the primary to the backup node.

Run `show checkpoint-data {process}`.

This command is also helpful in debugging synchronization problems that occur at run time.

The output displays, in percentages, the amount of copying completed by each process and the traffic statistics between the process on both the primary and the backup nodes.

Viewing Node Status

ExtremeXOS allows you to view node statistical information. Each node in a stackable switch in a SummitStack installed in your system is self-sufficient and runs the ExtremeXOS management applications. By reviewing this output, you can see the general health of the system along with other node parameters.

Run `show node {detail}`.

In a SummitStack, the `show stacking` command shows the node roles of active nodes.

Node Status Collected

Table 11 provides descriptions of node states.

Table 11: Node States

Node State	Description
BACKUP	In the backup state, this node becomes the primary node if the primary fails or enters the DOWN state. The backup node also receives the checkpoint state data from the primary.
DOWN	In the down state, the node is not available to participate in leader election. The node enters this state during any user action, other than a failure, that makes the node unavailable for management. Examples of user actions are: • Upgrading the softwareRebooting the system using the `reboot` command. • Initiating a failover using the `run failover` command. • Synchronizing the node software and configuration in non-volatile storage using the `synchronize` command.
FAIL	In the fail state, the node has failed and needs to be restarted or repaired. The node reaches this state if the system has a hardware or software failure.
INIT	In the initial state, the node is being initialized. A node stays in this state when it is coming up and remains in this state until it has been fully initialized. Being fully initialized means that all of the hardware has been initialized correctly and there are no diagnostic faults.
MASTER	In the primary (master) state, the node is responsible for all switch management functions.
STANDBY	In the standby state, leader election occurs—the primary and backup nodes are elected. The priority of the node is only significant in the standby state. In SummitStack, there can be more than two master-capable nodes. All such nodes that do not get elected either master or backup remain in standby state.

Understanding Hitless Failover Support

SummitStack node has customer ports that are under the control of its single central processor. In a SummitStack, failure of the primary node results in all ports that require

that node's processor for normal operation going down. The remaining SummitStack nodes' ports continue to function normally.

As described in the section Understanding System Redundancy on page 84, if you configure at least two master-capable nodes in a SummitStack, one assumes the role of primary and the other assumes the role of backup.

The primary node provides all of the switch management functions including bringing up and programming the other (standby) nodes in the SummitStack, running the bridging and routing protocols, and configuring the switch. The primary node also synchronizes the backup node in case it needs to take over the management functions if the primary node fails.

The configuration is one of the most important pieces of information checkpointed to the backup node. Each component of the system needs to checkpoint whatever runtime data is necessary to allow the backup node to take over as the primary node if a failover occurs, including the protocols and the hardware-dependent layers. For more information about checkpointing data and relaying configuration information, see Replicating Data Between Nodes on page 87.

Not all protocols support hitless failover. Layer 3 forwarding tables are maintained for pre-existing flows, but subsequent behavior depends on the routing protocols used. Static Layer 3 configurations and routes are hitless. You must configure *OSPF (Open Shortest Path First)* graceful restart for OSPF routes to be maintained, and you must configure *BGP (Border Gateway Protocol)* graceful restart for BGP routes to be maintained. For more information about OSPF, see OSPF and for more information about BGP, see BGP. For routing protocols that do not support hitless failover, the new primary node removes and re-adds the routes.

Protocol Support for Hitless Failover

Table 12 summarizes the protocol support for hitless failover. Unless otherwise noted, the behavior is the same for all switches.

If a protocol indicates support for hitless failover, additional information is also available in that particular chapter. For example, for information about network login support of hitless failover, see Network Login on page 1214.

Table 12: Protocol Support for Hitless Failover

Protocol	Behavior	Hitless
Bootstrap Protocol Relay	All bootprelay statistics (including option 82 statistics) are available on the backup node also	Yes
BGP	If you configure BGP graceful restart, by default the route manager does not delete BGP routes until 120 seconds after failover occurs. There is no traffic interruption. However, after BGP comes up after restart, BGP re-establishes sessions with its neighbors and relearns routes from all of them. This causes an increase in control traffic onto the network. If you do not configure graceful restart, the route manager deletes all BGP routes 1 second after the failover occurs, which results in a traffic interruption in addition to the increased control traffic.	Yes
Connectivity Fault Management (IEEE 802.1ag)	An ExtremeXOS process running on the master node should continuously send the MEP state changes to the backup. Replicating the protocol packets from an master node to a backup may be a huge overhead if CCMs are to be initiated/ received in the CPU and if the CCM interval is in the order of milliseconds. RMEP timeout does not occur on a remote node during the hitless failover. RMEP expiry time on the new master node in case of double failures, when the RMEP expiry timer is already in progress, is as follows: RMEP Expiry Time = elapsed expiry time on the master node + 3.5 * ccmIntervaltime + master node convergence time.	Yes
Dynamic Host Configuration Protocol client	The IP addresses learned on all *DHCP* enabled VLANs are retained on the backup node after failover.	Yes
Dynamic Host Configuration Protocol server	A DHCP server continues to maintain the IP addresses assigned to various clients and the lease times even after failover. When a failover happens, all the clients work as earlier.	Yes

Table 12: Protocol Support for Hitless Failover (continued)

Protocol	Behavior	Hitless
EAPS (Extreme Automatic Protection Switching)	The primary node replicates all EAPS BPDUs to the backup, which allows the backup to be aware of the state of the EAPS domain. Since both primary and backup nodes receive EAPS BPDUs, each node maintains equivalent EAPS states. By knowing the state of the EAPS domain, the EAPS process running on the backup node can quickly recover after a primary node failover. Although both primary and backup nodes receive EAPS BPDUs, only the primary transmits EAPS BPDUs to neighboring switches and actively participates in EAPS.	Yes
EDP (Extreme Discovery Protocol)	EDP does not checkpoint protocol data units (PDUs) or states, so the backup node does not have the neighbor's information. If the backup node becomes the primary node, and starts receiving PDUs, the new primary learns about its neighbors.	No
Extreme Loop Recovery Protocol (ELRP)	If you use ELRP as a standalone tool, hitless failover support is not needed since you initiate the loop detection. If you use ELRP in conjunction with *ESRP (Extreme Standby Router Protocol)*, ELRP does not interfere with the hitless failover support provided by ESRP. Although there is no hitless failover support in ELRP itself, ELRP does not affect the network behavior if a failover occurs.	No
Extreme Standby Router Protocol (ESRP)	If failover occurs on the ESRP MASTER switch, it sends a hello packet with the HOLD bit set. On receiving this packet, the ESRP SLAVE switch freezes all further state transitions. The MASTER switch keeps sending hellos with the HOLD bit set on every hello interval. When the MASTER is done with its failover, it sends another hello with the HOLD bit reset. The SLAVE switch resumes normal processing. (If no packet is received with the HOLD bit reset, the SLAVE timeouts after a certain time interval and resumes normal processing.) Failover on the ESRP SLAVE switch is of no importance because it is the SLAVE switch.	Yes

Table 12: Protocol Support for Hitless Failover (continued)

Protocol	Behavior	Hitless
Intermediate System-Intermediate System (IS-IS)	If you configure IS-IS graceful restart, there is no traffic interruption. However, after IS-IS comes up after restart, IS-IS re-establishes sessions with its neighbors and relearns Link State Packets (LSPs) from all of the neighbors. This causes an increase in network control traffic. If you do not configure graceful restart, the route manager deletes all IS-IS routes one second after the failover occurs, which results in a traffic interruption and increased control traffic. IS-IS for IPv6 does not support hitless restart .	IS-IS (IPv4) Yes IS-IS (IPv6) No
Link Aggregation Control Protocol (LACP)	If the backup node becomes the primary node, there is no traffic disruption.	Yes
LLDP (Link Layer Discovery Protocol)	LLDP is more of a tool than a protocol, so there is no hitless failover support. LLDP is similar to EDP, but there is also a MIB interface to query the information learned. After a failover, it takes 30 seconds or greater before the MIB database is fully populated again.	No
MSDP (Multicast Source Discovery Protocol)	If the master node fails, the MSDP process loses all state information and the backup node becomes active. However, the failover from the master to the backup causes MSDP to lose all state information and dynamic data, so it is not a hitless failover.	No
	All MLAG user configuration is executed on both master and backup nodes. Both nodes open listening health-check and checkpoint listening sockets on the respective well-known ports. All *FDB (forwarding database)* entries and IPMC group/cache information that were received through ISC checkpointing is synchronized to the backup node. After failover, the TCP session, which is handled by the failed master, tears down and there is a new session with the MLAG peer switch. After the failover, the FDB & McMgr processes trigger bulk checkpointing of all its entries to the MLAG peer upon receiving ISC up notification.	Yes

Table 12: Protocol Support for Hitless Failover (continued)

Protocol	Behavior	Hitless
Network Login	802.1X Authentication—Authenticated clients continue to remain authenticated after failover. However, one second after failover, all authenticated clients are forced to re-authenticate themselves. Information about unauthenticated clients is not checkpointed, so any such clients that were in the process of being authenticated at the instant of failover must go through the authentication process again from the beginning after failover. MAC-Based Authentication—Authenticated clients continue to remain authenticated after failover so the failover is transparent to them. Information about unauthenticated clients is not checkpointed so any such clients that were in the process of being authenticated at the instant of failover must go through the authentication process again from the beginning after failover. In the case of MAC-Based authentication, the authentication process is very short with only a single packet being sent to the switch so it is expected to be transparent to the client stations. Web-Based Authentication—Web-based Netlogin users continue to be authenticated after a failover.	Yes Yes Yes
OSPF	If you configure OSPF graceful restart, there is no traffic interruption. However, after OSPF comes up after restart, OSPF re-establishes sessions with its neighbors and relearns Link State Advertisements (LSAs) from all of the neighbors. This causes an increase in control traffic onto the network. If you do not configure graceful restart, the route manager deletes all OSPF routes one second after the failover occurs, which results in a traffic interruption in addition to the increased control traffic.	Yes

Table 12: Protocol Support for Hitless Failover (continued)

Protocol	Behavior	Hitless
OSPFv3 (Open Shortest Path First version 3)	If you configure OSPFv3 graceful restart, there is no traffic interruption. However, after OSPFv3 comes up after restart, OSPFv3 re-establishes sessions with its neighbors and relearns Link State Advertisements (LSAs) from all of the neighbors. If you do not configure graceful restart, the route manager deletes all OSPFv3 routes after failover occurs, which results in a traffic interruption in addition to the increased control traffic.	Yes
PoE (Power over Ethernet)	The PoE configuration is checkpointed to the backup node. This ensures that, if the backup takes over, all ports currently powered stay powered after the failover and the configured power policies are still in place.	Yes
Protocol Independent Multicast (PIM)	After a failover, all hardware and software caches are cleared and learning from the hardware is restarted. This causes a traffic interruption since it is the same as if the switch rebooted for all Layer 3 multicast traffic.	No
RIP (Routing Information Protocol)	RIP does not support graceful restart, so the route manager deletes all RIP routes one second after the failover occurs. This results in a traffic interruption as well as an increase in control traffic as RIP re-establishes its database.	No
RIPng (Routing Information Protocol Next Generation)	RIPng does not support graceful restart, so the route manager deletes all RIPng routes one second after the failover occurs. This results in a traffic interruption. After RIPng comes up on the new primary node, it relearns the routes from its neighbors. This causes an increase in control traffic onto the network.	No
Simple Network Time Protocol Client	*SNTP* client will keep the backup node updated about the last server from which a valid update was received, the time at which the last update was received, whether the SNTP time is currently good or not and all other statistics.	Yes

Table 12: Protocol Support for Hitless Failover (continued)

Protocol	Behavior	Hitless
STP (Spanning Tree Protocol)	STP supports hitless failover including catastrophic failure of the primary node without interruption. There should be no discernible network event external to the switch. The protocol runs in lock step on both master and backup nodes and the backup node is a hot spare that can take over at any time with no impact on the network.	Yes
VRRP (Virtual Router Redundancy Protocol)	VRRP supports hitless failover. The primary node replicates VRRP PDUs to the backup, which allows the primary and backup nodes to run VRRP in parallel. Although both nodes receive VRRP PDUs, only the primary transmits VRRP PDUs to neighboring switches and participates in VRRP.	Yes

Hitless Failover Caveats

This section describes the caveats for hitless failover.

Check the latest version of the ExtremeXOS release notes for additional information.

Caveats for a SummitStack

The following describes the hitless failover caveats for a SummitStack:

- All customer ports and the stacking links connected to the failed primary node will go down. In the recommended stack ring configuration, the stack becomes a daisy chain until the failed node restarts or is replaced.
- A brief traffic interruption (less than 50 milliseconds) can occur when the traffic on the ring is rerouted because the active topology becomes a daisy chain.
- Since the SummitStack can contain more than two master-capable nodes, it is possible to immediately elect a new backup node. If a new backup node is elected, when the original primary node restarts, it will become a standby node.
- To simulate the behavior of a chassis, a MAC address of one of the nodes is designated as the seed to form a stack MAC address. When a failover occurs, the SummitStack continues to be identified with this address.
- During an *OSPF* graceful restart, the SummitStack successfully restores the original link state database only if the OSPF network remains stable during the restart period. If the failed primary node provided interfaces to OSPF networks, the link state database restoration is prematurely terminated, and reconvergence occurs in the OSPF network due to the failover. See OSPF for a description of OSPF and the graceful restart function.
- During a *BGP* graceful restart, the SummitStack successfully restores the BGP routing table only if the BGP network remains stable during the restart period. If a receiving speaker detected the need for a routing change due to the failure of

links on the failed primary node, it deletes any previous updates it received from the restarting speaker (the SummitStack) before the restart occurred. Consequently, reconvergence occurs in the BGP network due to the failover. See BGP for a description of BGP and its graceful restart function.

Understanding Power Supply Management

Using Power Supplies

On ExtremeSwitching switches, ExtremeXOS reports when the PSU has power or has failed.

The ExtremeSwitching switches support an internal power supply with a range of 90V to 240V AC power as well as an external redundant power supply. The Extreme Networks External Power System (EPS) allows you to add a redundant power supply to the ExtremeSwitching switches to protect against a power supply failure. The EPS consists of a tray or module that holds the EPS power supplies.

Note
When an EPS-T tray with two EPS-160 PSUs is connected to a ExtremeSwitching switch, the internal power supply will show as failed.

On non-*PoE* ExtremeSwitching switches, if you experience an internal PSU failure and do not have an external PSU installed, the switch powers down. If you experience a PSU failure and have an external PSU installed, the switch uses the external PSU to maintain power to the switch.

On PoE ExtremeSwitching switches, there are specific power budget requirements and configurations associated with PoE that are not described in this section. The PoE ExtremeSwitching switches respond to internal and external PSU failures based on your PoE configurations. For more information about configuring PoE on the ExtremeSwitching PoE switches, see PoE.

For more information about ExtremeSwitching switches and EPS, see the hardware documentation listed in Related Publications.

Using Power Supplies--SummitStack Only

Since the nodes have their own power supplies and since they cannot be shared, management is the same as it is for standalone ExtremeSwitching switches.

The only difference is that the power management commands have been centralized so that they can be issued from the primary node.

Displaying Power Supply Information

Display the status of the currently installed power supplies on all switches.

1. Run `show power {ps_num} {detail}`.

   The detail option of this command shows power usage parameters on stacking and standalone ExtremeSwitching switches.
2. To view the system power status and the amount of available and required power, use the following command:

   ```
   show power budget
   ```
3. To display the status of the currently installed power supply controllers, use the following command:

   ```
   show power {ps_num}
   ```

Using the Network Time Protocol

Network Time Protocol (NTP) is used for synchronizing time on devices across a network with variable latency (time delay).

NTP provides a coordinated Universal Time Clock (UTC), the primary time standard by which the world regulates clocks and time. UTC is used by devices that rely on having a highly accurate, universally accepted time, and can synchronize computer clock times to a fraction of a millisecond. In a networked environment, having a universal time can be crucial. For example, the stock exchange and air traffic control use NTP to ensure accurate, timely data.

NTP uses a hierarchical, semi-layered system of levels of clock sources called a *stratum*. Each stratum is assigned a layer number starting with 0 (zero), with 0 meaning the least amount of delay. The stratum number defines the distance, or number of NTP hops away, from the reference clock. The lower the number, the closer the switch is to the reference clock. The stratum also serves to prevent cyclical dependencies in the hierarchy.

SNTP, as the name would suggest, is a simplified version of NTP that uses the same protocol, but without many of the complex synchronization algorithms used by NTP. SNTP is suited for use in smaller, less complex networks. For more information about SNTP see the section, Using the Simple Network Time Protocol on page 126.

Note
When NTP is enabled, it is not recommended to update the time manually.

Limitations

The Extreme Networks implementation of NTP includes the following limitations:

- cannot be enabled at the same time NTP is enabled.
- The NTP multicast delivery mechanism is not supported.
- The NTP autokey security mechanism is not supported.
- The broadcast client option cannot be enabled on a per- basis.

- NTP authentication can be enabled globally. It cannot be enabled per VR.
- VRF is not supported.
- The IPv6 address of a server received through DNS is not supported. The server's IPv6 address should be configured directly.
- Restrict list is not supported for IPv6.
- Broadcast client is not supported for IPv6.

VR Configuration Support

Starting with ExtremeXOS 22.2, NTP can be configured over multiple VRs.

To enable NTP on a VR, use the following command:

```
enable ntp vr vr_name
```

Note
If VR is not specified, command is executed in current VR context and enables NTP for that VR alone.

To configure NTP over VLANs, use the following command:

```
enable ntp vlan vlan-name [broadcast-server | key keyid]
```

To configure NTP in all VLANs for a VR, use the following command:

```
enable ntp vr vr_name
```

The VR configuration can be seen in `show ntp {{vr}vr_name}` command output.

Configuring the NTP Server/Client

An NTP server provides clock information to NTP or *SNTP* clients. You can configure an NTP server as an NTP client to receive clock information from more reliable external NTP servers or a local clock. You can also build a hierarchical time distribution topology by using TCP/IP. The switch can work as both an NTP client and server at the same time to build a hierarchical clock distribution tree. This hierarchical structure eliminates the need for a centralized clock server and provides a highly available clock tree with minimal network load and overhead.

Beginning in version 32.6, NTP servers and peers can be configured in IPv6 address format. Additionally, when enabling NTP on a VLAN, NTP is enabled on all the global IPv6 addresses of the VLAN. You can also have a combination of IPv4 and IPv6 servers or peers.

- To configure an NTP server, use the following command:
  `configure ntp [server | peer] add [ip_address | ipv6_address | host_name] {key keyid} {option [burst | initial-burst]} {{vr} vr_name}`
- To delete an NTP server, use the following command:
  `configure ntp [server | peer] delete [ip_address | ipv6_address | host_name]`

- To display NTP server information, use the following commands:

  ```
  show ntp server
  ```

With scaled configuration, NTP takes more time to apply the configuration, since it involves system calls to apply the configuration.

Configuring Restrict Lists

You can configure a restrict list to deny/permit a particular host or network. The following limitations apply:

- You cannot add an NTP server to the restrict list. When the NTP server is added, the server IP address is added as the system restrict list with the action "permit".
- If an NTP is enabled on a *VLAN*, that VLAN IP address/network cannot be added as restrict list entry.

To create or delete a restrict list, use the following command:

```
configure ntp restrict-list [add | delete] network {mask} [permit |
deny] {{vr} vr_name}
```

To show the NTP access list of the current system based on the source IP address blocks, use the following command:

```
show ntp restrict-list {user | system} {{vr}vr_name}
```

Managing NTP Peer Support

An NTP peer is a member of a group of NTP servers. Normally, an NTP peer is used to synchronize clock information among a group of servers that serve as mutual backups for each other. Typically, core switches are configured as NTP peers, and an NTP server is configured as a core switch to an NTP client, aggregation switch, or edge switch. An NTP client can choose the most reliable clock from all servers that have a peer relationship with the client.

- To configure an NTP peer, use the following command:

  ```
  configure ntp [server | peer] add [ip_address | ipv6_address |
  host_name] {key keyid} {option [burst | initial-burst]} {{vr} vr_name}
  ```

- To delete an NTP peer, use the following command:

  ```
  configure ntp [server | peer] delete [ip_address | ipv6_address |
  host_name]
  ```

- To display an NTP peer, use the following command:

  ```
  show ntp
  show ntp association [{ip_address} | {host_name}] {{vr} vr_name}
  show ntp association [{ip_address} | {host_name}] statistics
  {{vr}vr_name}
  show ntp sys-info
  ```

Managing NTP Local Clock Support

A local clock serves as backup to distribute clock information internally when reliable external clock sources are not reachable. Assign a higher stratum value to the local clock to ensure that it is not selected when an external reliable clock source with a lower stratum number exists.

- To configure a local clock, use the following command:

  ```
  configure ntp local-clock stratum stratum_number
  ```

- To delete a local clock, use the following command:

  ```
  configure ntp local-clock none
  ```

Managing NTP Broadcast Server Support

An NTP broadcast server sends periodic time updates to a broadcast address in a LAN. When a broadcast client is configured for NTP, that client can receive time information from the broadcasted NTP packets. Using broadcast packets can greatly reduce the NTP traffic on a network, especially in a network with many NTP clients.

To ensure that NTP broadcast clients get clock information from the correct NTP broadcast servers, with minimized risks from malicious NTP broadcast attacks, configure RSA Data Security, Inc. *MD5 (Message-Digest algorithm 5)* Message-Digest Algorithm authentication on both the NTP broadcast server and NTP clients.

- To configure an NTP broadcast server over a *VLAN* where NTP broadcast service is provided, use the following command:

  ```
  enable ntp {vlan} vlan-name broadcast-server {key keyid}
  ```

- To delete an NTP broadcast server over a VLAN where NTP broadcast service is enabled, use the following command:

  ```
  disable ntp {vlan} vlan-name broadcast-server
  ```

- To display an NTP broadcast server, use the following command:

  ```
  show ntp server
  ```

  ```
  show ntp vlan {{vr} vr_name}
  ```

Managing NTP Broadcast Client Support

An NTP client listens for NTP packets from an NTP broadcast server. To listen for network broadcast messages, enable an NTP broadcast client. This option is global (it cannot be enabled on a per-*VLAN* basis).

- To configure an NTP broadcast client, use the following command:

  ```
  enable ntp broadcast-client {{vr} vr_name}
  ```

- To delete an NTP broadcast client, use the following command:

  ```
  disable ntp broadcast-client {{vr} vr_name}
  ```

- To display an NTP broadcast client, use the following command:

  ```
  show ntp {{vr}vr_name}
  ```

Managing NTP Authentication

To prevent false time information from unauthorized servers, enable NTP authentication to allow an authenticated server and client to exchange time information. The currently supported authentication methods are the RSA Data Security, Inc. *MD5* Message-Digest Algorithm and SHA-256.

When FIPS mode is enabled, Network Time Protocol (NTP) uses OpenSSL Federal Information Processing Standards (FIPS) library and supports only FIPS-compliant algorithms for authentication (SHA-256 authentication only). MD5 key configuration support is not available when FIPS mode is enabled, and existing MD5 key configurations are removed when FIPS mode comes into effect. For more information about FIPS mode, see Federal Information Processing Standards (FIPS) Mode on page 1533.

First, enable NTP authentication globally on the switch. Then create an NTP authentication key configured as trusted, to check the encryption key against the key on the receiving device before an NTP packet is sent. After configuration is complete, an NTP server, peer, and broadcast server can use NTP authenticated service.

- To enable or disable NTP authentication globally on the switch, use the following command:

  ```
  enable ntp authentication
  ```

  ```
  disable ntp authentication
  ```

- To create or delete an RSA Data Security, Inc. MD5 Message-Digest Algorithm key for NTP authentication, use the following command:

  ```
  create ntp key keyid [md5 | sha256] {encrypted encrypted_key_string | key_string}
  ```

  ```
  delete ntp key [keyid | all]
  ```

- To configure an RSA Data Security, Inc. MD5 Message-Digest Algorithm key as trusted or not trusted, use the following command:

  ```
  configure ntp key keyid [trusted | not-trusted]
  ```

- To display RSA Data Security, Inc. MD5 Message-Digest Algorithm authentication, use the following command:

  ```
  show ntp key
  ```

- To display NTP authentication, use the following command:

  ```
  show ntp sys-info
  ```

  If NTP authentication is enabled, then "Authentication" flag is set in "System Flags" output.

Showing NTP Information

To display the global NTP status of the switch, use the command:

```
show ntp
```

To display the current system status based on the most reliable clock server or NTP server, use the following command:

```
show ntp sys-info
```

To display NTP *VLAN* configurations, use the command:

```
show ntp vlan {{vr} vr_name}
```

To display NTP VR configurations, use the command:

```
show ntp {{vr}vr_name}
```

To display all of the NTP clock source information, from a statically configured server, peer, or broadcast server:

```
show ntp association [{ip_address} | {host_name}] {{vr} vr_name}
```

NTP Configuration Example

In the example shown in Figure 3, SW#1 synchronizes its clock from the 0-3.us.pool.ntp.org timer server, and provides the synchronized clock information to SW#2 as a unicast message, and to SW#3 as a broadcast message.

SW#2 configures SW#1 as a time server using a normal unicast message. It also has a local clock (127.127.1.1) with a stratum level of 10. SW#3 is configured as broadcast client without specific server information. For security purposes, SW#2 and SW#3 use RSA Data Security, Inc. *MD5* Message-Digest Algorithm authentication with a key index of 100.

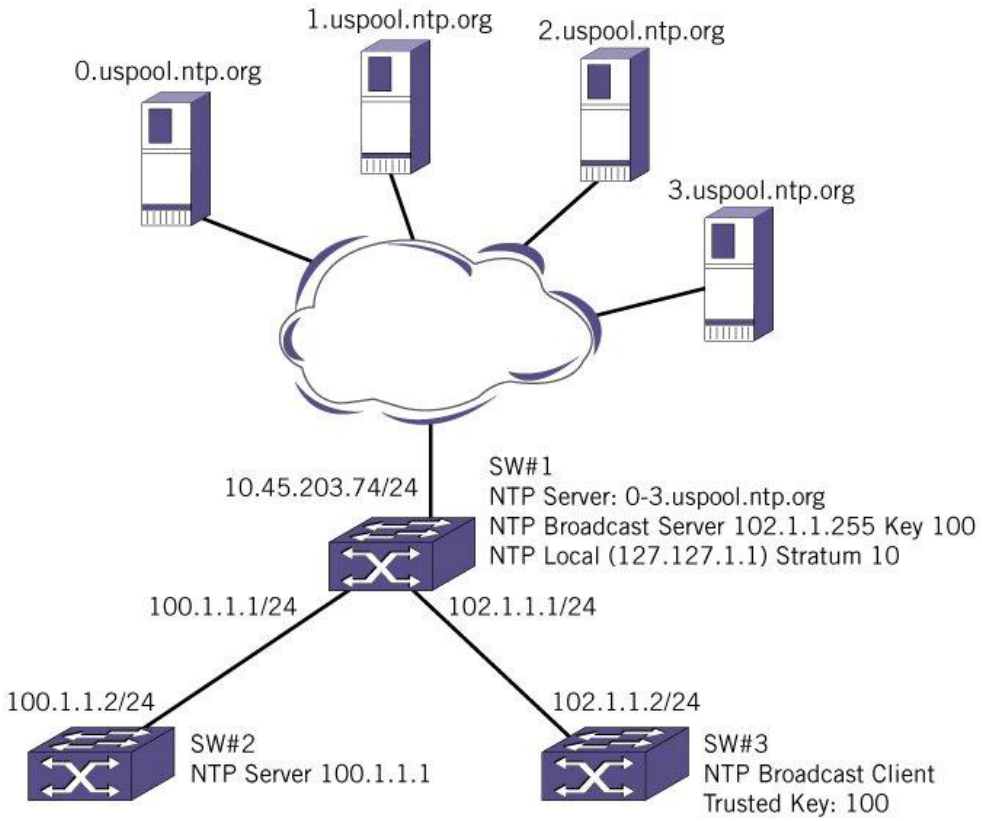


Figure 3: NTP Configuration Example

SW#1 Configuration

```
create vlan internet
create vlan toSW2
create vlan toSW3
config vlan internet add port 1
config vlan toSW2 add port 2
config vlan toSW3 add port 3
config vlan internet ipaddress 10.45.203.74/24
config vlan toSW2 ipaddress 100.1.1.1/24
config vlan toSW3 ipaddress 102.1.1.1/24
config iproute add default 10.45.203.1 vr vr-default
enable ntp
create ntp key 100  md5 EXTREME
config ntp key 100 trusted
enable ntp vlan internet
enable ntp vlan toSW2
enable ntp vlan toSW3
enable ntp vlan toSW3 broadcast-server key 100
config ntp server add 0.us.pool.ntp.org
config ntp server add 1.us.pool.ntp.org
config ntp server add 2.us.pool.ntp.org
config ntp server add 3.us.pool.ntp.org
config ntp local-clock stratum 10
config ntp server add 1000::1 vr VR-Default
```

SW#2 Configuration

```
create vlan toSW1
config vlan toSW1 add port 1
config vlan toSW1 ipaddress 100.1.1.2/24
enable ntp
enable ntp vlan toSW1
config ntp server add 100.1.1.1
```

SW#3 Configuration

```
create vlan toSW1
config vlan toSW1 add port 1
config vlan toSW1 ipaddress 102.1.1.2/24
enable ntp
enable ntp broadcast-client
create ntp key 100 md5 EXTREME
config ntp key 100 trusted
enable ntp vlan toSW1
```

NTP Configuration Example over Multiple VRs

Switch SW#1 Configuration

```
create vlan v2 -- in vr VR-Mgmt
create vlan v5 -- in vr VR-Default
create vlan v3 -- in vr test (user-VR)
create vlan v6 -- in vr test (user-VR)
config vlan v2 ipaddress 10.127.7.33/24
config vlan v5 ipaddress 5.1.1.5/24
config vlan v3 ipaddress 10.127.7.60/24
config vlan v5 add port <port towards SW#2>
config vlan v6 add port <port towards SW#3>
config vlan v2 add port <port connected to reach reference clock>
config vlan v3 add port <port connected to reach reference clock>
enable ntp vr test
enable ntp vr VR-Default
enable ntp vr VR-Mgmt
enable ntp vlan v3
enable ntp vlan v6
enable ntp vlan v2
enable ntp vlan v5
enable ntp vlan Mgmt
configure ntp server add 128.138.140.44 initial-burst vr VR-Default
configure ntp server add 128.138.141.172 initial-burst vr VR-Default
configure ntp server add 129.6.15.28 initial-burst vr test
configure ntp server add 129.6.15.29 initial-burst vr test
configure ntp server add 129.6.15.30 initial-burst vr test
configure ntp server add 2.us.pool.ntp.org initial-burst vr test
configure ntp server add 204.152.184.72 initial-burst vr VR-Mgmt
configure ntp server add 24.56.178.140 initial-burst vr VR-Default
configure ntp server add 64.113.32.5 initial-burst vr VR-Default
configure ntp server add 66.199.22.67 initial-burst vr VR-Mgmt
configure ntp server add 98.175.203.200 initial-burst vr VR-Mgmt
```

SW#2 Configuration

```
create vlan v5 -- in vr VR-Default
config vlan v5 add port <port towards SW#1>
config vlan v5 ipaddress 5.1.1.6/24
enable ntp vr VR-Default
enable ntp vlan v5
configure ntp server add 5.1.1.5 initial-burst vr VR-Default
```

SW#3 Configuration

```
create vlan v6 -- in vr test (user-VR)
config vlan v6 ipaddress 6.1.1.7/24
config vlan v6 add port <port towards SW#1>
enable ntp vr test
enable ntp vlan v6
configure ntp server add 6.1.1.6 initial-burst vr test
```

Using the Simple Network Management Protocol

Any network manager program running the *SNMP* can manage the switch if the Management Information Base (MIB) is installed correctly on the management station.

Each network manager program provides its own user interface to the management facilities.

Note
When using a network manager program to create a *VLAN*, we do not support the SNMP createAndWait operation. To create a VLAN with SNMP, use the createAndGo operation.

createAndGo is one of six values in the RowStatus column of SMIv2 tables. createAndGo is supplied by a manager wishing to create a new instance of a conceptual row and have its status automatically set to active in order to make it available for use by the managed device

The following sections describe how to get started if you want to use an SNMP manager. It assumes you are already familiar with SNMP management.

Note
Perform a save operation if you make any configurations using SNMP mibs. If you do not save, some of the configurations may not survive when you reboot.

Enabling and Disabling SNMPv2c and SNMPv3

ExtremeXOS can concurrently support SNMPv2c and SNMPv3. The default is that both types of *SNMP* are disabled. Network managers can access the device with either SNMPv2c methods or SNMPv3.

- To allow support for all SNMP access, or SNMPv2c access only, or SNMPv3 access only, use the following command:
  `enable snmp access {snmp-v2c | snmpv3}`
- To prevent support for all SNMP access, or SNMPv2c access only, or SNMPv3 access only, use the following command:
  `disable snmp access {snmp-v2c | snmpv3}`

Most of the commands that support SNMPv2c use the keyword snmp; most of the commands that support SNMPv3 use the keyword snmpv3.

After a switch reboot, all slots must be in the “Operational” state before SNMP can manage and access the slots. To verify the current state of the slot, use the `show slot` command.

Understanding Safe Defaults Mode and SNMP

The safe defaults mode runs an interactive script that allows you to enable or disable *SNMP*, Telnet, and switch ports.

When you set up your switch for the first time, you must connect to the console port to access the switch. After logging in to the switch, you enter safe defaults mode. Although Telnet and switch ports are enabled by default, the script prompts you to confirm those settings or disable them.

SNMP is disabled by default. If you choose to enable SNMP, the switch follows the interactive script asking you if you want to enable SNMPv2c and/or SNMPv3.

For more detailed information about safe defaults mode, see Using Safe Defaults Mode on page 48.

Enabling and Disabling SNMP Access on Virtual Routers

Beginning with ExtremeXOS 12.4.2 software, you can enable and disable *SNMP* access on any or all VRs. By default, SNMP access is enabled on all VRs.

SNMP access for a VR has global SNMP status that includes all SNMPv2c, SNMPv3, and default group status. However, trap receiver configuration and trap enabling/disabling are independent of global SNMP access and are still forwarded on a VR that is disabled for SNMP access.

- Enable SNMP access on a VR:
  `enable snmp access vr [vr_name | all]`
- Disable SNMP access on a VR, use the following command:
  `disable snmp access vr [vr_name | all]`
- Display the SNMP configuration and statistics on a VR:
  `show snmp {vr} vr_name`

SNMP Authentication Traps

You can enable/disable SNMP authentication traps that synchronizes changes between the **show switch management** command and SNMP GET. To enable SNMP authentication traps, use the following command:

```
enable snmp authentication-traps
```

To disable SNMP authentication traps, use the following command:

```
disable snmp authentication-traps
```

Use the **show switch management** command to display SNMP authentication trap configuration information.

Accessing Switch Agents

To access the *SNMP* agent residing in the switch, at least one *VLAN* must have an assigned IP address. ExtremeXOS supports either IPv4 or IPv6 addresses to manage the switch.

By default, SNMP access and SNMPv2c traps are enabled. SNMP access and SNMP traps can be disabled and enabled independently—you can disable SNMP access but still allow SNMP traps to be sent, or vice versa.

Return-to-Normal SNMP Notifications

This feature implements two new *SNMP* notifications that indicate that an alert condition has “returned-to-normal”. The first notification addresses CPU utilization. Currently, ExtremeXOS allows you to monitor the CPU utilization and history for all of the processes running on the switch. When this function is enabled, a CPU threshold value is used to flag a process in the system that has exceeded the threshold. A SNMP notification is generated for processes exceeding that threshold. When a process’ cpu utilization falls back below the configured threshold, this feature adds support to generated a new “return-to-normal” notification.

The second notification is a “return-to-normal” message that corresponds to a previously generated overheat notification. The overheat notification indicates that the on board temperature sensor has reported a overheat condition. When the on board temperature sensor reports the clearing of an overheat condition, the new “return-to-normal” notification is generated.

Supported MIBs

Standard MIBs supported by the switch. In addition to private MIBs, the switch supports the standard MIBs listed in Supported Standards, Protocols, and MIBs on page 1854.

Configuring SNMPv2c Settings

The following SNMPv2c parameters can be configured on the switch:

- **Authorized trap receivers**—An authorized trap receiver can be one or more network management stations on your network. The switch sends SNMPv2c traps to all configured trap receivers. You can specify a community string and UDP port individually for each trap receiver. All community strings must also be added to the switch using the `configure snmp add community` command.

  To configure a trap receiver on a switch, use the following command:

  Note
  It is recommended to use a different community name for the trap receiver and other *SNMP* read/write operations.

  ```
  configure snmp add trapreceiver [ip_address | ipv6_address] community
  [[hex hex_community_name] | community_name] {port port_number} {from
  [src_ip_address | src_ipv6_address]} {vr vr_name} {mode trap_mode}
  ```

To configure the notification type (trap/inform), use the following command specifying trap as the type:

```
configure snmpv3 add notify [[hex hex_notify_name] | notify_name] tag
[[hex hex_tag] | tag] {type [trap | inform]}{volatile}
```

To delete a trap receiver on a switch, use the following command:

```
configure snmp delete trapreceiver [[ip_address | ipv6_address]
{port_number} | all]
```

- SNMP INFORM—SNMP INFORM allows for confirmation of a message delivery. When an SNMP manager receives an INFORM message from an SNMP agent, it sends a confirmation response back to the agent. If the message has not been received and therefore no response is returned, the INFORM message is resent. You can configure the number of attempts to make and the interval between attempts.

  To configure the notification type (trap/inform), use the following command specifying inform as the type:

  ```
  configure snmpv3 add notify [[hex hex_notify_name] | notify_name] tag
  [[hex hex_tag] | tag] {type [trap | inform]}{volatile}
  ```

  To configure the number of SNMP INFORM notification retries, use the following command:

  ```
  configure snmpv3 target-addr [[hex hex_addr_name] | addr_name] retry
  retry_count
  ```

  To configure the SNMP INFORM timeout interval, use the following command:

  ```
  configure snmpv3 target-addr [[hex hex_addr_name] | addr_name] timeout
  timeout_val
  ```

- Community strings—The community strings allow a simple method of authentication between the switch and the remote network manager. There are two types of community strings on the switch:
  - Read community strings provide read-only access to the switch.
  - Read-write community strings provide read- and-write access to the switch.

  To store and display the SNMP community string in encrypted format, use the following command:

  ```
  configure snmpv3 add community [[hex hex_community_index] |
  community_index] [encrypted name community_name | name [[hex
  hex_community_name] | community_name] {store-encrypted} ] user
  [[hex hex_user_name] | user_name] {tag [[hex transport_tag] |
  transport_tag]} {volatile}
  ```

  Note
  SNMP community string name can contain special characters.

- System contact (optional)—The system contact is a text field that enables you to enter the name of the person(s) responsible for managing the switch.

- **System name** (optional)—The system name enables you to enter a name that you have assigned to this switch. The default name is the model name of the switch (for example, BD-1.2).
- **System location** (optional)—Using the system location field, you can enter the location of the switch.

Displaying SNMP Settings

To view *SNMP* settings configured on the switch, use the following command:

```
show switch management
```

This command displays the following information:

- Enable/disable state for Telnet and SNMP access
- Login statistics
- Enable/disable state for idle timeouts
- Maximum number of CLI sessions
- SNMP community strings
- SNMP notification type (trap or INFORM)
- SNMP trap receiver list
- SNMP trap receiver source IP address
- Authentication trap configuration
- SNMP statistics counter
- SSH access state (enabled/disabled).
- CLI configuration logging
- SNMP access states of v1, v2c disabled and v3 enabled
- Access-profile usage configured via ACLs for additional Telnet, SSH2 security, SNMP, and HTTP(s)
- CLI scripting settings
- Enable/disable state
- Error message setting
- Persistence mode
- Dropped SNMP packet counter

SNMPv3

SNMPv3 is an enhanced standard for *SNMP* that improves the security and privacy of SNMP access to managed devices and provides sophisticated control of access to the device MIB. The prior standard versions of SNMP, SNMPv1, and SNMPv2c, provided no privacy and little security.

The following RFCs provide the foundation for the Extreme Networks implementation of SNMPv3:

- RFC 3410, Introduction to version 3 of the Internet-standard Network Management Framework, provides an overview of SNMPv3.
- RFC 3411, An Architecture for Describing SNMP Management Frameworks, talks about SNMP architecture, especially the architecture for security and administration.
- RFC 3412, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), talks about the message processing models and dispatching that can be a part of an SNMP engine.
- RFC 3413, SNMPv3 Applications, talks about the different types of applications that can be associated with an SNMPv3 engine.
- RFC 3414, The User-Based Security Model for Version 3 of the Simple Network Management Protocol (SNMPv3), describes the User-Based Security Model (USM).
- RFC 3415, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), talks about VACM as a way to access the MIB.
- RFC 3826, The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model.

Note
AES 192 and AES 256 bit encryption are proprietary implementations and may not work with some SNMP Managers.

The SNMPv3 standards for network management were driven primarily by the need for greater security and access control. The new standards use a modular design and model management information by cleanly defining a message processing (MP) subsystem, a security subsystem, and an access control subsystem.

The MP subsystem helps identify the MP model to be used when processing a received Protocol Data Unit (PDU), which are the packets used by SNMP for communication.

The MP layer helps in implementing a multilingual agent, so that various versions of SNMP can coexist simultaneously in the same network.

The security subsystem features the use of various authentication and privacy protocols with various timeliness checking and engine clock synchronization schemes.

SNMPv3 is designed to be secure against:

- Modification of information, where an in-transit message is altered.
- Masquerades, where an unauthorized entity assumes the identity of an authorized entity.
- Message stream modification, where packets are delayed and/or replayed.
- Disclosure, where packet exchanges are sniffed (examined) and information is learned about the contents.

The access control subsystem provides the ability to configure whether access to a managed object in a local MIB is allowed for a remote principal. The access control scheme allows you to define access policies based on MIB views, groups, and multiple security levels.

In addition, the SNMPv3 target and notification MIBs provide a more procedural approach for generating and filtering of notifications.

SNMPv3 objects are stored in non-volatile memory unless specifically assigned to volatile storage. Objects defined as permanent cannot be deleted.

Note
In SNMPv3, many objects can be identified by a human-readable string or by a string of hexadecimal octets. In many commands, you can use either a character string, or a colon-separated string of hexadecimal octets to specify objects. To indicate hexadecimal octets, use the keyword hex in the command.

Message Processing

A particular network manager may require messages that conform to a particular version of *SNMP*. The choice of the SNMPv1, SNMPv2c, or SNMPv3 MP model can be configured for each network manager as its target address is configured. The selection of the MP model is configured with the **mp-model** keyword in the following command:

```
configure snmpv3 add target-params [[hex hex_param_name] |
param_name ]user [[hex hex_user_name] | user_name] mp-model [snmpv2c |
snmpv3] sec-model [snmpv2c | usm] {sec-level [noauth | authnopriv |
priv]} {volatile}
```

SNMPv3 Security

In SNMPv3 the User-Based Security Model (USM) for *SNMP* was introduced. USM deals with security related aspects like authentication, encryption of SNMP messages, and defining users and their various access security levels. This standard also encompasses protection against message delay and message replay.

USM Timeliness Mechanisms

An Extreme Networks switch has one SNMPv3 engine, identified by its snmpEngineID. The first four octets are fixed to 80:00:07:7C, which represents the Extreme Networks vendor ID. By default, the additional octets for the snmpEngineID are generated from the device MAC address.

Every SNMPv3 engine necessarily maintains two objects: SNMPEngineBoots, which is the number of reboots the agent has experienced and SNMPEngineTime, which is the local time since the engine reboot. The engine has a local copy of these objects and the latestReceivedEngineTime for every authoritative engine it wants to communicate with. Comparing these objects with the values received in messages and then applying certain rules to decide upon the message validity accomplish protection against message delay or message replay.

In a SummitStack, the MAC address chosen for the snmpEngineID is the configured stack MAC address.

Configuring USM Timeliness Mechanism

Configure the snmpEngineID and SNMPEngineBoots from the command line. The snmpEngineID can be configured from the command line.

SNMPEngineBoots can be set to any desired value but will latch on its maximum, 2147483647.

1. To set the snmpEngineID, use the following command:

   ```
   configure snmpv3 engine-id hex_engine_id
   ```

2. To set the SNMPEngineBoots, use the following command:

   ```
   configure snmpv3 engine-boots (1-2147483647)
   ```

Users, Groups, and Security

SNMPv3 controls access and security using the concepts of users, groups, security models, and security levels.

Users are created by specifying a user name. Depending on whether the user will be using authentication and/or privacy, you would also specify an authentication protocol (RSA Data Security, Inc. *MD5* Message-Digest Algorithm or SHA) with password or key, and/or privacy (DES, AES) password or key.

Managing Users

Users are created by specifying a user name. Enabling the SNMPv3 user access allows an end user to access the MIBs using SNMPv3 user. By deleting users access, the end-user is not able to access the switch/MIBs using SNMPv3 user.

- To create a user, use the following command:

  ```
  configure snmpv3 add user [[hex hex_user_name] | user_name]
  {authentication [md5 | sha] [hex hex_auth_password | auth_password]}
  {privacy {des | aes {128 | 192 | 256}} [[hex hex_priv_password] |
  priv_password]} }{volatile}
  ```

- To display information about a user, or all users, use the following command:

  ```
  show snmpv3 user {[[hex hex_user_name] | user_name]}
  ```

- To delete a user, use the following command:

  ```
  configure snmpv3 delete user [all | [[hex hex_user_name] | user_name]
  { }]
  ```

  Note
  The SNMPv3 specifications describe the concept of a security name. In the ExtremeXOS implementation, the user name and security name are identical. In this manual, both terms are used to refer to the same thing.

- To configure SNMPv3 password policies for all users, use the following command:

  ```
  configure snmpv3 user password-policy [authentication | privacy]
  [[username-match | char-repeat | long-sequence | char-validation]
  ```

[**permit** | **deny**] | **history** [*num_passwords* | **duration** *days* | **none**] | **min-length** [*num_characters* | **none**]]

- To display information about configured password policies, use the following command:

  ```
  show snmpv3 user password-policy
  ```

SNMPv3 Extended Password Length Support

Version 33.6.1 extends SNMPv3 password length support from 48 to 128 characters, enabling compliance with stringent security requirements.

Password Length Enhancement

The maximum password length for SNMPv3 user authentication and privacy passwords has been increased from 48 to 128 characters. Combined with the existing password policy commands, administrators can now enforce minimum password lengths up to 128 characters to meet advanced security requirements while maintaining backward compatibility with existing configurations.

Extended password length applies to:

- **Authentication passwords:** MD5 and SHA authentication (8-128 characters, previously 8-48)
- **Privacy passwords:** DES, 3DES, and AES encryption (8-128 characters, previously 8-48)

Password formats supported:

- **ASCII:** Standard text passwords
- **Hexadecimal:** Passwords specified in hexadecimal format
- **Localized keys:** Pre-computed authentication and privacy keys

Creating SNMPv3 Users with Extended Passwords

The syntax for creating SNMPv3 users remains unchanged. The extended password length range is available automatically:

```
configure snmpv3 add user <user_name>
    authentication [md5 | sha] <auth_password>
    privacy [des | 3des | aes {128 | 192 | 192-legacy | 256 | 256-legacy}] <priv_password>
```

Example with 128-character passwords:

```
configure snmpv3 add user secureuser authentication md5

"12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890123456789012345678"
    privacy aes 128

"12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890123456789012345678"
```

Note
Passwords are stored and displayed in encrypted form for security. The above example shows the clear-text password during initial configuration only.

Password Policy Configuration

The password policy commands allow you to enforce minimum password lengths and other security requirements. The minimum length enforcement has been extended to support the new 128-character maximum.

Configuring Minimum Password Length:

Set minimum password length for authentication passwords:

```
configure snmpv3 user password-policy authentication min-length <num_characters>
```

Set minimum password length for privacy passwords:

```
configure snmpv3 user password-policy privacy min-length <num_characters>
```

Where *num_characters* is between 8 and 128 (default 8).

Example enforcing 64-character minimum passwords:

```
configure snmpv3 user password-policy authentication min-length 64
configure snmpv3 user password-policy privacy min-length 64
```

To disable minimum length enforcement:

```
configure snmpv3 user password-policy authentication min-length none
configure snmpv3 user password-policy privacy min-length none
```

Password Policy Enforcement

When minimum length policies are configured, password creation and modification is validated against the policy. Passwords that do not meet the minimum length requirement are rejected with an error message.

Example of policy enforcement:

```
# Set 64-character minimum
configure snmpv3 user password-policy authentication min-length 64

# Attempt to create user with short password fails
configure snmpv3 add user testuser authentication md5 "password1234567890"
Error: Password length cannot be less than 64 characters.

# User creation succeeds with compliant password
configure snmpv3 add user testuser authentication md5
    "12345678901234567890123456789012345678901234567890123456789012345"
# (success - no error message)
```

Policy enforcement applies to:

- Creating new SNMPv3 users
- Modifying existing user passwords
- Both authentication and privacy passwords independently

Additional Password Policy Options

The SNMPv3 password policy supports several security requirements beyond minimum length. These options remain unchanged and work with the extended password length:

Username Match Prevention:

```
configure snmpv3 user password-policy authentication username-match deny
configure snmpv3 user password-policy privacy username-match deny
```

Prevents passwords from matching the username.

Character Repeat Prevention:

```
configure snmpv3 user password-policy authentication char-repeat deny
configure snmpv3 user password-policy privacy char-repeat deny
```

Prevents passwords containing the same letter or number twice in a row (e.g., "aa", "11").

Long Sequence Prevention:

```
configure snmpv3 user password-policy authentication long-sequence deny
configure snmpv3 user password-policy privacy long-sequence deny
```

Prevents passwords containing sequences of more than 3 characters in alphabetical, numerical, or keyboard order (e.g., "qwer", "1234").

Character Validation:

```
configure snmpv3 user password-policy authentication char-validation deny
configure snmpv3 user password-policy privacy char-validation deny
```

Requires passwords to contain at least one each of: uppercase letters, lowercase letters, numbers, and special characters.

Password History:

```
configure snmpv3 user password-policy authentication history <num_passwords>
configure snmpv3 user password-policy privacy history <num_passwords>
```

Or:

```
configure snmpv3 user password-policy authentication history duration <days>
configure snmpv3 user password-policy privacy history duration <days>
```

Prevents reuse of recent passwords (1-10 previous passwords, or passwords used within 1-365 days).

To permit any policy (return to default):

```
configure snmpv3 user password-policy authentication username-match permit
```

To disable a policy:

```
configure snmpv3 user password-policy authentication history none
```

Viewing Password Policy Configuration

To view current SNMPv3 password policy settings, use the **show snmpv3** command. The output displays configured policy requirements including minimum length:

```
show snmpv3
```

Sample output excerpt:

```
SNMPv3 Password Policy:
  Authentication:
    Minimum Length    : 64 characters
    Username Match    : Deny
    Character Repeat  : Deny
    Long Sequence     : Deny
    Character Validation : Deny
    History           : Last 5 passwords
  Privacy:
    Minimum Length    : 64 characters
    Username Match    : Deny
    Character Repeat  : Deny
    Long Sequence     : Deny
    Character Validation : Deny
    History           : Last 5 passwords
```

Backward Compatibility

Existing SNMPv3 user configurations with passwords up to 48 characters continue to function normally. The extended password length is optional and enforced only when explicitly configured via password policy commands.

If minimum length is not configured or set below the existing password length, existing users are not affected. Only new user creation and password modifications are subject to the minimum length requirement.

Example:

- Existing user "admin" has a 32-character password
- Administrator sets minimum length to 64 characters
- User "admin" continues to authenticate successfully with existing password
- When changing "admin" password, new password must meet 64-character minimum
- New users must have passwords meeting 64-character minimum

Security Considerations

Longer passwords enhance security but require careful consideration:

Benefits:

- Greater resistance to brute-force attacks
- Increased entropy and cryptographic strength
- Compliance with stringent security policies and regulations
- Better protection against dictionary attacks

Operational Considerations:

- Very long passwords are harder to type manually and more prone to typos
- Consider using password managers for generating and storing complex passwords
- Document password requirements for SNMP managers and monitoring systems

- Test SNMP manager compatibility with extended password lengths before deployment
- Plan password rotation procedures for long passwords

Best Practices:

- Balance security requirements with operational practicality
- For highly secure environments, use 64+ character passwords
- For standard deployments, 20-40 character passwords provide good security
- Combine minimum length with other password policy requirements (character validation, history)
- Use different passwords for authentication and privacy
- Rotate passwords regularly according to your security policy
- Store passwords securely in password management systems

Hexadecimal and Localized Key Passwords

Extended password length also applies to hexadecimal and localized key formats:

Hexadecimal Passwords:

```
configure snmpv3 add user hexuser authentication md5 hex <hex_auth_password>
    privacy aes 128 hex <hex_priv_password>
```

Hexadecimal passwords allow specifying passwords as hexadecimal strings. The byte length limits still apply (8-128 bytes), but each byte is represented by two hexadecimal characters, so the hex string length is 16-256 characters.

Localized Keys:

```
configure snmpv3 add user keyuser authentication md5 localized-key <auth_localized_key>
    privacy aes 128 localized-key <priv_localized_key>
```

Localized keys are pre-computed MD5 or SHA hashes of the engine-id and user's password. Using localized keys avoids sending clear-text passwords over the network and can be more secure in certain deployment scenarios.

Integration with SNMP Managers

When configuring SNMP managers to use extended passwords:

1. Verify your SNMP manager software supports passwords up to 128 characters
2. Configure the SNMPv3 user on the switch with the desired password length
3. Configure the same username and passwords in your SNMP manager
4. Test connectivity using SNMPv3 commands:
   ```
   snmpget -v3 -u secureuser -l authPriv -a MD5 -A "yourpassword" -x AES -X
   "yourpassword"
       <switch-ip> sysDescr.0
   ```
5. Monitor for authentication failures that may indicate password mismatch or length issues

If your SNMP manager does not support extended password lengths, you can continue using passwords up to 48 characters without any issues.

Use Case Example

A financial institution requires all passwords to meet a 64-character minimum to comply with security regulations:

```
# Step 1: Configure password policy
configure snmpv3 user password-policy authentication min-length 64
configure snmpv3 user password-policy authentication char-validation deny
configure snmpv3 user password-policy authentication history 10
configure snmpv3 user password-policy privacy min-length 64
configure snmpv3 user password-policy privacy char-validation deny
configure snmpv3 user password-policy privacy history 10

# Step 2: Create SNMPv3 user with compliant password
# (Use a password manager to generate a strong 64+ character password)
configure snmpv3 add user monitor_user authentication sha
    "Mk9@p3L#wQ7xZ$2nR8vY4tB6hD1fG5jK3mP0sA9uN7cV2eX6wQ8yT4rL1oI5hG3fD7sA9pZ0"
    privacy aes 256
    "wQ8pL3@xZ7#nR2$vY4tB6hD1fG5jK9mP0sA3uN7cV2eX6wQ8yT4rL1oI5hG3fD7sA9pZ0Mk9@"

# Step 3: Verify user creation
show snmpv3 user

# Step 4: Configure SNMP manager with same credentials
# Step 5: Test connectivity
```

Managing Groups

Groups are used to manage access for the MIB. You use groups to define the security model, the security level, and the portion of the MIB that members of the group can read or write.

The security model and security level are discussed in Security Models and Levels on page 120. The view names associated with a group define a subset of the MIB (subtree) that can be accessed by members of the group. The read view defines the subtree that can be read, write view defines the subtree that can be written to, and notify view defines the subtree that notifications can originate from. MIB views are discussed in Setting SNMPv3 MIB Access Control on page 121.

A number of default groups are already defined. These groups are: admin, initial, v2c_ro, v2c_rw.

Enabling SNMPv3 default-group access activates the access to an SNMPv3 default group and the user- created SNMPv3-user part of default group.

Disabling SNMPv3 default-group access removes access to users and user created users who are part of the default-group.

Note
SNMPv3 default group should be enabled by default. If it is disabled snmpv2 community will not work.

The user-created authenticated SNMPv3 users (who are part of a user-created group) are able to access the switch.

- To underscore the access function of groups, groups are defined using the following command:

  ```
  configure snmpv3 add access [[hex hex_group_name] | group_name] {sec-model [snmpv2c | usm]} {sec-level [noauth | authnopriv | priv]} {read-view [[hex hex_read_view_name] | read_view_name]} {write-view [[hex hex_write_view_name]] | write_view_name]} {notify-view [[hex hex_notify_view_nam]] | notify_view_name]} {volatile}
  ```

- To display information about the access configuration of a group or all groups, use the following command:

  ```
  show snmpv3 access {[[hex hex_group_name] | group_name]}
  ```

- To enable default-group, use the following command:

  ```
  enable snmpv3 default-group
  ```

- To disable a default-group, use the following command:

  ```
  disable snmpv3 default-group
  ```

- To associate users with groups, use the following command:

  ```
  configure snmpv3 add group [[hex hex_group_name] | group_name] user [[hex hex_user_name] | user_name] {sec-model [snmpv2c | usm]} {volatile}
  ```

- To show which users are associated with a group, use the following command:

  ```
  show snmpv3 group {[[hex hex_group_name] | group_name] {user [[hex hex_user_name] | user_name]}}
  ```

- To delete a group, use the following command:

  ```
  configure snmpv3 delete access [all-non-defaults | {[[hex hex_group_name] | group_name] {sec-model [snmpv2c | usm] sec-level [noauth | authnopriv | priv]}}]
  ```

  When you delete a group, you do not remove the association between the group and users of the group.

- To delete the association between a user and a group, use the following command:

  ```
  configure snmpv3 delete group {[[hex hex_group_name] | group_name]} user [all-non-defaults | {[[hex hex_user_name] | user_name] {sec-model [snmpv2c|usm]}}]
  ```

Security Models and Levels

For compatibility, SNMPv3 supports three security models:

- SNMPv2c—community strings-based security
- SNMPv3—USM security

The default is USM. You can select the security model based on your network manager.

The three security levels supported by USM are:

- noAuthnoPriv—No authentication, no privacy. This is the case with existing SNMPv2c agents.

- AuthnoPriv—Authentication, no privacy. Messages are tested only for authentication.
- AuthPriv—Authentication, privacy. This represents the highest level of security and requires every message exchange to pass the authentication and encryption tests.

When a user is created, an authentication method is selected, and the authentication and privacy passwords or keys are entered.

When RSA Data Security, Inc. *MD5* Message-Digest Algorithm authentication is specified, HMAC-MD5-96 is used to achieve authentication with a 16-octet key, which generates a 128-bit authorization code. This authorization code is inserted in the msgAuthenticationParameters field of SNMPv3 PDUs when the security level is specified as either AuthnoPriv or AuthPriv. Specifying SHA authentication uses the HMAC-SHA protocol with a 20-octet key for authentication.

For privacy, the user can select any one of the following supported privacy protocols: DES, AES 128/192/256. In the case of DES, a 16-octet key is provided as input to DES-CBS encryption protocol which generates an encrypted PDU to be transmitted. DES uses bytes 1-7 to make a 56 bit key. This key (encrypted itself) is placed in msgPrivacyParameters of SNMPv3 PDUs when the security level is specified as AuthPriv.

The *SNMP* Context Name should be set to the name for which the information is requested. If the Context Name is not set the switch will retrieve the information for "*VR-Default*". If the SNMP request is targeted for the protocols running per (see Adding and Deleting Routing Protocols on page 2208), then the contextName should be set to the exact virtual-Router for which the information is requested. List of protocols running per Virtual Router:

- *BGP*
- *OSPF*
- PIM
- *RIP*
- *OSPFv3*
- RIPNG
- *MPLS (Multiprotocol Label Switching)*
- ISIS

Setting SNMPv3 MIB Access Control

SNMPv3 provides a fine-grained mechanism for defining which parts of the MIB can be accessed. This is referred to as the View-Based Access Control Model (VACM).

MIB views represent the basic building blocks of VACM. They are used to define a subset of the information in the MIB. Access to read, to write, and to generate notifications is based on the relationship between a MIB view and an access group. The users of the access group can then read, write, or receive notifications from the part of the MIB defined in the MIB view as configured in the access group.

A view name, a MIB subtree/mask, and an inclusion or exclusion define every MIB view. For example, there is a System group defined under the MIB-2 tree. The Object Identifier (OID) for MIB-2 is 1.3.6.1.2, and the System group is defined as MIB-2.1.1, or directly as 1.3.6.1.2.1.1.

When you create the MIB view, you can choose to include the MIB subtree/mask or to exclude the MIB subtree/mask.

In addition to the user-created MIB views, there are three default views: defaultUserView, defaultAdminView, and defaultNotifyView.

MIB views that are used by security groups cannot be deleted.

- To define a MIB view which includes only the System group, use the following subtree/mask combination:

  1.3.6.1.2.1.1/1.1.1.1.1.1.1.0

  The mask can also be expressed in hex notation (used in the ExtremeXOS CLI):

  1.3.6.1.2.1.1/fe
- To define a view that includes the entire MIB-2, use the following subtree/mask:

  1.3.6.1.2.1.1/1.1.1.1.1.0.0.0

  which, in the CLI, is:

  1.3.6.1.2.1.1/f8
- To create a MIB view, use the following command:

  `configure snmpv3 add mib-view [[hex hex_view_name] | view_name] subtree object_identifier {subtree_mask} {type [included | excluded]} {volatile}`

  After the view has been created, you can repeatedly use the `configure snmpv3 add mib-view` command to include and/or exclude MIB subtree/mask combinations to precisely define the items you want to control access to.
- To show MIB views, use the following command:

  `show snmpv3 mib-view {[[hex hex_view_name] | view_name] {subtree object_identifier}}`
- To delete a MIB view, use the following command:

  `configure snmpv3 delete mib-view [all-non-defaults | {[[hex hex_view_name] | view_name] {subtree object_identifier}}]`

SNMPv3 Notification

SNMPv3 can use SNMPv2c notifications to send information from an agent to the network manager.

The terms *trap* and *notification* are used interchangeably in this context. Notifications are messages sent from an agent to the network manager, typically in response to some state change on the agent system. With SNMPv3, you can define precisely

which traps you want sent, to which receiver by defining filter profiles to use for the notification receivers.

To configure notifications, you configure a target address for the target that receives the notification, a target parameters name, and a list of notification tags. The target parameters specify the security and MP models to use for the notifications to the target. The target parameters name also points to the filter profile used to filter the notifications. Finally, the notification tags are added to a notification table so that any target addresses using that tag will receive notifications.

Configuring Target Addresses

A target address is similar to the earlier concept of a trap receiver.

- To configure a target address, use the following command:

  `configure snmpv3 add target-addr [[hex hex_addr_name] | addr_name] param [[hex hex_param_name] |param_name ] ipaddress [ ip_address | ipv6_address ] {transport-port port_number} {from [src_ip_address | src_ipv6_address]} {vr vr_name} {tag-list [tag_list | hex hex_tag_list]} {volatile}`

  In configuring the target address you supply an address name that identifies the target address, a parameters name that indicates the MP model and security for the messages sent to that target address, and the IP address and port for the receiver. The parameters name also is used to indicate the filter profile used for notifications.

  The from option sets the source IP address in the notification packets.

  The tag-list option allows you to associate a list of tags with the target address. The tag defaultNotify is set by default.
- To display target addresses, use the following command:

  `show snmpv3 target-addr {[[hex hex_addr_name] | addr_name]}`
- To delete a single target address or all target addresses, use the following command:

  `configure snmpv3 delete target-addr [{[[hex hex_addr_name] | addr_name]} | all]`

Managing Notification Tags

When you create a target address, either you associate a list of notification tags with the target or by default, the defaultNotify tag is associated with the target. When the system generates notifications, only those targets associated with tags currently in the standard MIB table, called snmpNotifyTable, are notified.

Note
This notification entry can be deleted via CLI and also via MIB. If this is deleted, then this can result in the traps not being sent for trap receivers which do not have the tag-list value mentioned explicitly.

- To add an entry to the table, use the following command:

  `configure snmpv3 add notify [[hex hex_notify_name] | notify_name] tag [[hex hex_tag] | tag] {type [trap | inform]}{volatile}`

Any targets associated with tags in the snmpNotifyTable are notified, based on the filter profile associated with the target.

- To display the notifications that are set, use the following command:
  `show snmpv3 notify {[[hex hex_notify_name] | notify_name]}`
- To delete an entry from the snmpNotifyTable, use the following command:
  `configure snmpv3 delete notify [{[[hex hex_notify_name] | notify_name]} | all-non-defaults]`

Configuring Notifications

Because the target parameters name points to a number of objects used for notifications, configure the target parameter name entry first.

You can then configure the target address, filter profiles and filters, and any necessary notification tags.

Access Profile Logging for SNMP

The access profile logging feature allows you to use an *ACL* policy file or dynamic ACL rules to control access to *SNMP* services on the switch.

When access profile logging is enabled for SNMP, the switch logs messages and increments counters when packets are denied access to SNMP. No messages are logged for permitted access.

You can manage SNMP access using one (not both) of the following methods:

- Create and apply an ACL policy file.
- Define and apply individual ACL rules.

One advantage of ACL policy files is that you can copy the file and use it on other switches. One advantage to applying individual ACL rules is that you can enter the rules at the CLI command prompt, which can be easier than opening, editing, and saving a policy file.

ACL Match Conditions and Actions

The ACLs section describes how to create *ACL* policies and rules using match conditions and actions. Access profile logging supports the following match conditions and actions:

- Match conditions
  - Source-address—IPv4 and IPv6
- Actions
  - Permit
  - Deny

If the ACL is created with more match conditions or actions, only those listed above are used for validating the packets. All other conditions and actions are ignored.

The source-address field allows you to identify an IPv4 address, IPv6 address, or subnet mask for which access is either permitted or denied.

If the traffic does not match any of the rules, the default behavior is deny.

Limitations

Access profile logging for SNMP has the following limitations:

- Either policy files or ACL rules can be associated with SNMP, but not both at the same time.
- Only source-address match is supported.
- Access-lists that are associated with one or more applications (SNMP or Telnet, for example) cannot be directly deleted. They must be unconfigured from the application first and then deleted from the CLI.
- Default counter support is added only for ACL rules and not for policy files.

Managing ACL Policies for SNMP

The ACLs section describes how to create *ACL* policy files.

- Configure *SNMP* to use an ACL policy using one of the following commands:

  ```
  configure snmp access-profile profile_name

  configure snmp access-profile profile_name readwrite
  ```

  By default, SNMP supports the readwrite option to change the current configuration.
- To configure SNMP to remove a previously configured ACL policy, use the following command:

  ```
  configure snmp access-profile none
  ```

Managing ACL Rules for SNMP

Before you can assign an *ACL* rule to SNMP, you must create a dynamic ACL rule as described in ACLs.

- To add or delete a rule for *SNMP* access, use the following command:

  ```
  configure snmp access-profile [ access_profile {readonly | readwrite}
  | [[add rule ] [first | [[before | after] previous_rule]]] | delete
  rule | none ]
  ```

  Note
  The readonly option acts like the readwrite option when it is enabled.

- To display the access-list permit and deny statistics for an application, use the following command:

  ```
  show access-list counters process [snmp | telnet | ssh2 | http]
  ```

Misconfiguration Error Messages

The following messages can appear during configuration of policies or rules for the SNMP service:

Rule <rule> is already applied	A rule with the same name is already applied to this service.

Please remove the policy <policy> already configured, and then add rule <rule>	A policy file is already associated with the service. You must remove the policy before you can add a rule.
Rule <previous_rule> is not already applied	The specified rule has not been applied to the service, so you cannot add a rule in relation to that rule.
Rule <rule> is not applied	The specified rule has not been applied to the service, so you cannot remove the rule from the service.
Error: Please remove previously configured rule(s) before configuring policy <policy>	A policy or one or more ACL rules are configured for the service. You must delete the remove the policy or rules from the service before you can add a policy.

Using the Simple Network Time Protocol

ExtremeXOS supports the client portion of the *SNTP* Version 3 based on *RFC4330*.

SNTP can be used by the switch to update and synchronize its internal clock from a Network Time Protocol (NTP) server. After SNTP has been enabled, the switch sends out a periodic query to the indicated NTP server, or the switch listens to broadcast NTP updates. In addition, the switch supports the configured setting for Greenwich Mean time (GMT) offset and the use of Daylight Saving Time.

Configuring and Using SNTP

To use *SNTP*:

1. Identify the host(s) that are configured as NTP server(s). Additionally, identify the preferred method for obtaining NTP updates. The options are for the NTP server to send out broadcasts, or for switches using NTP to query the NTP server(s) directly. A combination of both methods is possible. You must identify the method that should be used for the switch being configured.
2. Configure the Greenwich Mean Time (GMT) offset and Daylight Saving Time preference. The command syntax to configure GMT offset and usage of Daylight Saving Time is as follows:

   ```
   configure timezone {name tz_name} GMT_offset {autodst {name
   dst_timezone_ID} {dst_offset} begins [every floatingday | on
   absoluteday] {at time_of_day_hour time_of_day_minutes} {ends
   [every floatingday | on absoluteday] {at time_of_day_hour
   time_of_day_minutes}}}
   ```

   By default beginning in 2007, Daylight Saving Time is assumed to begin on the second Sunday in March at 2:00 AM, and end the first Sunday in November at 2:00 AM and to be offset from standard time by one hour.

   a. If this is the case in your time zone, you can set up automatic daylight saving adjustment with the command:

      ```
      configure timezone GMT_offset autodst
      ```

b. If your time zone uses starting and ending dates and times that differ from the default, you can specify the starting and ending date and time in terms of a floating day, as follows:

```
configure timezone name MET 60 autodst name MDT begins every last sunday march
at 1 30 ends every last sunday october at 1 30
```

c. You can also specify a specific date and time, as shown in the following command:

```
configure timezone name NZST 720 autodst name NZDT 60 begins every first sunday
october
at 2 00 ends on 3 16 2004 at 2 00
```

The optional time zone IDs are used to identify the time zone in display commands such as `show switch {detail}`.

Table 13 describes the time zone command options in detail.

Table 13: Time Zone Configuration Command Options

tz_name	Specifies an optional name for this timezone specification. May be up to six characters in length. The default is an empty string.
GMT_offset	Specifies a Greenwich Mean Time (GMT) offset, in + or - minutes.
autodst	Enables automatic Daylight Saving Time.
dst_timezone_ID	Specifies an optional name for this Daylight Saving Time specification. May be up to six characters in length. The default is an empty string.
dst_offset	Specifies an offset from standard time, in minutes. Value is from 1–60. The default is 60 minutes.
floatingday	Specifies the day, week, and month of the year to begin or end Daylight Saving Time each year. Format is *week day month* where: • *week* is specified as [first | second | third | fourth | last] • *day* is specified as [sunday | monday | tuesday | wednesday | thursday | friday | saturday] • *month* is specified as [january | february | march | april | may | june | july | august | september | october | november | december] Default for beginning is second sunday march; default for ending is first sunday november.
absoluteday	Specifies a specific day of a specific year on which to begin or end DST. Format is *month day year* where: • *month* is specified as 1–12 • *day* is specified as 1–31 • *year* is specified as 1970–2035 The year must be the same for the begin and end dates.
time_of_day_hour	Specifies the time of day to begin or end Daylight Saving Time. May be specified as an hour (0–23). The default is 2.
time_of_day_minutes	Specify the minute to begin or end Daylight Saving Time. May be specified as a minute (0–59).
noautodst	Disables automatic Daylight Saving Time.

3. Automatic Daylight Saving Time changes can be enabled or disabled.

   The default setting is enabled. To disable automatic Daylight Saving Time, use the command:

   `configure timezone {name tz_name} GMT_offset noautodst`

4. Enable the SNTP client using the following command:

   `enable sntp-client`

   After SNTP has been enabled, the switch sends out a periodic query to the NTP servers defined in the next step (if configured) or listens to broadcast NTP updates from the network. The network time information is automatically saved into the onboard real-time clock.

5. If you would like this switch to use a directed query to the NTP server, configure the switch to use the NTP server(s). An NTP server can be an IPv4 address or an IPv6 address or a hostname. If the switch listens to NTP broadcasts, skip this step. To configure the switch to use a directed query, use the following command:

   `configure sntp-client [primary | secondary] host-name-or-ip {vr vr_name}`

   The following two examples use an IPv6 address as an NTP server and a hostname as an NTP server:

   ```
   configure sntp-client primary fd98:d3e2:f0fe:0:54ae:34ff:fecc:892
   configure sntp-client primary ntpserver.mydomain.com
   ```

   NTP queries are first sent to the primary server. If the primary server does not respond within one second, or if it is not synchronized, the switch queries the secondary server (if one is configured). If the switch cannot obtain the time, it restarts the query process. Otherwise, the switch waits for the sntp-client update interval before querying again.

6. Optionally, the interval for which the SNTP client updates the real-time clock of the switch can be changed using the following command:

   `configure sntp-client update-interval update-interval`

   The default sntp-client update-interval value is 64 seconds.

7. Verify the configuration.

   a. `show sntp-client`

      This command provides configuration and statistics associated with SNTP and its connectivity to the NTP server.

   b. `show switch {detail}`

      This command indicates the GMT offset, the Daylight Saving Time configuration and status, and the current local time.

NTP updates are distributed using GMT time.

To properly display the local time in logs and other time-stamp information, the switch should be configured with the appropriate GMT offset to GMT based on geographical location.

GMT Offsets

Table 14 lists offsets for GMT.

Table 14: Greenwich Mean Time Offsets

GMT Offset in Hours	GMT Offset in Minutes	Common Time Zone References	Cities
+0:00	+0	GMT - Greenwich Mean UT or UTC - Universal (Coordinated) WET - Western European	London, England; Dublin, Ireland; Edinburgh, Scotland; Lisbon, Portugal; Reykjavik, Iceland; Casablanca, Morocco
-1:00	-60	WAT - West Africa	Cape Verde Islands
-2:00	-120	AT - Azores	Azores
-3:00	-180		Brasilia, Brazil; Buenos Aires, Argentina; Georgetown, Guyana
-4:00	-240	AST - Atlantic Standard	Caracas; La Paz
-5:00	-300	EST - Eastern Standard	Bogota, Columbia; Lima, Peru; New York, NY, Trevor City, MI USA
-6:00	-360	CST - Central Standard	Mexico City, Mexico
-7:00	-420	MST - Mountain Standard	Saskatchewan, Canada
-8:00	-480	PST - Pacific Standard	Los Angeles, CA, Santa Clara, CA, Seattle, WA USA
-9:00	-540	YST - Yukon Standard	
-10:00	-600	AHST - Alaska-Hawaii Standard CAT - Central Alaska HST - Hawaii Standard	
-11:00	-660	NT - Nome	
-12:00	-720	IDLW - International Date Line West	
+1:00	+60	CET - Central European FWT - French Winter MET - Middle European MEWT - Middle European Winter SWT - Swedish Winter	Paris France; Berlin, Germany; Amsterdam, The Netherlands; Brussels, Belgium; Vienna, Austria; Madrid, Spain; Rome, Italy; Bern, Switzerland; Stockholm, Sweden; Oslo, Norway
+ 2:00	+120	EET - Eastern European, Russia Zone 1	Athens, Greece; Helsinki, Finland; Istanbul, Turkey; Jerusalem, Israel; Harare, Zimbabwe
+3:00	+180	BT - Baghdad, Russia Zone 2	Kuwait; Nairobi, Kenya; Riyadh, Saudi Arabia; Moscow, Russia; Tehran, Iran

Table 14: Greenwich Mean Time Offsets (continued)

GMT Offset in Hours	GMT Offset in Minutes	Common Time Zone References	Cities
+4:00	+240	ZP4 - Russia Zone 3	Abu Dhabi, UAE; Muscat; Tblisi; Volgograd; Kabul
+5:00	+300	ZP5 - Russia Zone 4	
+5:30	+330	IST - India Standard Time	New Delhi, Pune, Allahabad, India
+6:00	+360	ZP6 - Russia Zone 5	
+7:00	+420	WAST - West Australian Standard	
+8:00	+480	CCT - China Coast, Russia Zone 7	
+9:00	+540	JST - Japan Standard, Russia Zone 8	
+10:00	+600	EAST - East Australian Standard GST - Guam Standard Russia Zone 9	
+11:00	+660		
+12:00	+720	IDLE - International Date Line East NZST - New Zealand Standard NZT - New Zealand	Wellington, New Zealand; Fiji, Marshall Islands
+13:00	+780	PHOT-Phoenix Island Time	Kanton Island
+14:00	+840	LINT-Line Islands Time	Kiritimati

SNTP Example

In this example, the switch queries a specific NTP server and a backup NTP server.

The switch is located in Cupertino, California, and an update occurs every 20 minutes. The commands to configure the switch are as follows:

```
configure timezone -480 autodst
configure sntp-client update-interval 1200
enable sntp-client
configure sntp-client primary 10.0.1.1
configure sntp-client secondary 10.0.1.2
```

Access Profile Logging for HTTP/HTTPS

The access profile logging feature allows you to use dynamic *ACL* rules to control access to Hypertext Transfer Protocol (HTTP) services on the switch.

When access profile logging is enabled for HTTP, the switch logs messages and increments counters when packets are denied access to HTTP. No messages are logged for permitted access.

Note
For more information on ExtremeXOS software support for HTTP, see Hypertext Transfer Protocol on page 1643.

You can manage HTTP access using one (not both) of the following methods:

- Create and apply an ACL policy file
- Define and apply individual ACL rules

One advantage of ACL policy files is that you can copy the file and use it on other switches. One advantage to applying individual ACL rules is that you can enter the rules at the CLI command prompt, which can be easier than opening, editing, and saving a policy file.

ACL Match Conditions and Actions

The ACLs section describes how to create *ACL* policies and rules using match conditions and actions. Access profile logging supports the following match conditions and actions:

- Match conditions
  - Source-address—IPv4 and IPv6
- Actions
  - Permit
  - Deny

If the ACL is created with more match conditions or actions, only those listed above are used for validating the packets. All other conditions and actions are ignored.

The source-address field allows you to identify an IPv4 address, IPv6 address, or subnet mask for which access is either permitted or denied.

If the HTTP traffic does not match any of the rules, the default behavior is deny.

Limitations

Access profile logging for HTTP/HTTPS has the following limitations:

- Policy file support is not available for HTTP and HTTPS.
- Only source-address match is supported.
- Access-lists that are associated with one or more applications cannot be directly deleted. They must be unconfigured from the application first and then deleted from the CLI.

Managing ACL Rules for HTTP

Before you can assign an *ACL* rule to HTTP, you must create a dynamic ACL rule as described in ACLs.

- To add or delete a rule for HTTP access, use the following command:

  `configure web http access-profile [[[`**`add`** *`rule`* `] [`**`first`** `| [[`**`before`** `| `**`after`**`] `*`previous_rule`*`]]] | `**`delete`** *`rule`* `| `**`none`** `]`

- To display the access-list permit and deny statistics for an application, use the following command:

  `show access-list counters process [`**`snmp`** `| `**`telnet`** `| `**`ssh2`** `| `**`http`**`]`

Misconfiguration Error Messages

The following messages can appear during configuration of policies or rules for the SNMP service:

Rule <rule> is already applied	A rule with the same name is already applied to this service.
Please remove the policy <policy> already configured, and then add rule <rule>	A policy file is already associated with the service. You must remove the policy before you can add a rule.
Rule <previous_rule> is not already applied	The specified rule has not been applied to the service, so you cannot add a rule in relation to that rule.
Rule <rule> is not applied	The specified rule has not been applied to the service, so you cannot remove the rule from the service.
Error: Please remove previously configured rule(s) before configuring policy <policy>	A policy or one or more ACL rules are configured for the service. You must delete the remove the policy or rules from the service before you can add a policy.

Enabling and Disabling Switch Management Access

Version 33.1.1 adds support for enabling and disabling all management access modes in a switch. The following command enables management access:

```
enable switch access
```

Once this command is used, the following commands are also run internally:

- `enable ssh2`
- `enable telnet`
- `enable web http`
- `enable web https`
- `enable snmp access`

Only these commands are present in the configuration once they finish executing. This configuration must be saved.

Similarly, you can use the following command to disable all switch management access:

```
disable switch access
```

The following commands also run as a result:

- `disable ssh2` (also disables SFTP)
- `disable telnet`
- `disable web {http | https}` (also disables REST-API)
- `disable snmp access`

Because this disables all management access, any existing sessions are closed and only console connection can be used. The following warning displays to confirm your action:

```
# disable switch access
Warning: This command will disable access to switch via SSH, Telnet, HTTP, HTTPS and
SNMP. Existing such sessions will be logged out and user can access the switch only using
console connection.
Continue? (y/N) Yes
```

Managing the ExtremeXOS Software

Using the ExtremeXOS File System on page 135
Managing the Configuration File on page 139
Managing ExtremeXOS Processes on page 140
Understanding Memory Protection on page 144

The software platform is a distributed software architecture.

The distributed architecture consists of separate binary images organized into discrete software modules with messaging between them. The software and system infrastructure subsystem form the basic framework of how the software applications interact with each other, including the system startup sequence, memory allocation, and error events handling. Redundancy and data replication is a built-in mechanism of the operating system. The system infrastructure provides basic redundancy support and libraries for all of the software applications.

Note
For information about downloading and upgrading a new software image, saving configuration changes, and upgrading the BootROM, see Software Upgrade and Boot Options on page 1697.

Like any advanced operating system, the operating system gives you the tools to manage your switch and create your network configurations.

The following enhancements and functionality have been added to the switch operating system:

- File system administration
- Configuration file management
- Process control
- Memory protection

File system administration

With the enhanced file system, you can move, copy, and delete files from the switch. The file system structure allows you to keep, save, rename, and maintain multiple copies of configuration files on the switch. In addition, you can manage other entities of the switch such as policies and access control lists (ACLs).

Configuration file management

With the enhanced configuration file management, you can oversee and manage multiple configuration files on your switch. In addition, you can upload, download, modify, and name configuration files used by the switch.

Process control

With process control, you can stop and start processes, restart failed processes, and update the software for a specific process or set of processes.

Memory protection

With memory protection, each function can be bundled into a single application module running as a memory protected process under real-time scheduling. In essence, the software protects each process from every other process in the system. If one process experiences a memory fault, that process cannot affect the memory space of another process.

The following sections describe in more detail how to manage the software.

Using the ExtremeXOS File System

The file system in ExtremeXOS is the structure by which files are organized, stored, and named.

The switch can store multiple user-defined configuration and policy files, each with its own name. Using a series of commands, you can manage the files on your system. For example, you can rename or copy a configuration file on the switch, display a comprehensive list of the configuration and policy files on the switch, or delete a policy file from the switch.

Note
File names are case-sensitive. For information on file name restrictions, see the specific command in the Switch Engine v33.6.1 Command References .

You can also download configuration and policy files from the switch to a network Trivial File Transfer Protocol (TFTP) server using TFTP. For detailed information about downloading switch configurations, see Software Upgrade and Boot Options on page 1697. For detailed information about downloading policies and ACLs, see ACLs on page 341.

With guidance from Extreme Networks Technical Support personnel, you can configure the switch to capture core dump files, which contain debugging information that is useful in troubleshooting situations. For more information about configuring core dump files and managing the core dump files stored on your switch, see Understanding Core Dump Messages on page 1710.

File are stored at:

- USB: `/usr/local/ext`
- Internal memory: `/usr/local/tmp`
- Home directory: `/usr/local/cfg`

The command `ls` without specifying the path returns results for just the home directory.

By default, core dumps are stored in the internal memory space (`/usr/local/tmp`), so to see them, run the command `ls /usr/local/tmp`.

List Files

```
# ls /usr/local/tmp
-rw-r--r-- 1 root root 7429 Sep 22 16:19 core.nvram.1
-rw-r--r-- 1 root root 7100 Sep 28 08:40 core.nvram.2
drwxrwxrwx 2 root root 1024 Sep 28 08:43 dhcp
-rw-rw-rw- 1 root root 1072 Sep 28 08:43 trigger_log_slot1.txt
```

TFTP Files

```
# tftp put 10.6.48.39 /usr/local/tmp/core.nvram.1 mmitchell/core.nvram.1
Uploading core.nvram.1 to 10.6.48.39 ... done!
```

Copy Files

```
# cp /usr/local/tmp/core.nvram.1 core.copy
Copy 'core.nvram.1' from '/usr/local/tmp' to '/usr/local/cfg/core.copy'? (y/N)
```

Move Files

```
# mv /usr/local/tmp/core.nvram.1 core.moveToHome
Move 'core.nvram.1' from '/usr/local/tmp' to '/usr/local/cfg/core.moveToHome'? (y/N)
```

Moving or Renaming Files on the Switch

You can move or rename an existing configuration, policy, or if configured, core dump file in the system.

XML-formatted configuration files have a .cfg file extension. The switch runs only .cfg files. ASCII-formatted configuration files have an .xsf file extension. See Uploading ASCII-Formatted Configuration Files on page 1715 for more information. Policy files have a .pol file extension.

When you rename a file, make sure the it uses the same file extension as the original file. If you change the file extensions, the file may be unrecognized by the system. For example, if you have an existing configuration file named test.cfg, the new filename must include the .cfg extension.

1. Run the `mv` command.

   ```
   # mv test.cfg megset.cfg
   Rename config test.cfg to config megtest.cfg on switch? (y/n)
   ```

2. Enter `y` to rename the file on your system. Enter `n` to cancel this process and keep the existing filename.

   If you attempt to rename an active configuration file (the configuration currently selected the boot the switch), the switch displays an error similar to the following:
   ```
   Error: Cannot rename current selected active configuration.
   ```

For more information about configuring core dump files and managing the core dump files stored on your switch, see Understanding Core Dump Messages on page 1710.

Copying Files on the Switch

The copy function allows you to make a copy of an existing file before you alter or edit the file. By making a copy, you can easily go back to the original file if needed.

XML-formatted configuration files have a .cfg file extension. The switch runs only .cfg files. ASCII-formatted configuration files have an .xsf file extension. See Uploading ASCII-Formatted Configuration Files on page 1715 for more information. Policy files have a .pol file extension.

When you copy a configuration or policy file from the system, make sure you specify the appropriate file extension. For example, if you want to copy a policy file, specify the filename and .pol.

1. Copy an existing configuration or policy file on your switch using the `cp` command.

   ```
   cp test.cfg test1.cfg
   Copy config test.cfg to config test1.cfg on switch? (y/n)
   ```

2. Enter `y` to copy the file. Enter `n` to cancel this process and not copy the file.

   When you enter `y`, the switch copies the file with the new name and keeps a backup of the original file with the original name. After the switch copies the file, use the `ls` command to display a complete list of files.

   Note
   If you make a copy of a file, such as a core dump file, you can easily compare new information with the old file if needed.

For more information about configuring the storage of core dump files, see Understanding Core Dump Messages on page 1710.

Displaying Files on the Switch

You can display a list of the configuration, policy, or if configured, core dump files stored on your switch by running the command `ls {file_name}`.

When you do not specify a parameter, this command lists all of the files in the current directory stored on your switch.

If you do specify a parameter, you can refer to a specific directory to view all of the files in that directory.

Output from this command includes the file size, date and time the file was last modified, and the file name.

For more information about configuring core dump files and managing the core dump files stored on your switch, see Understanding Core Dump Messages on page 1710.

Transferring Files to and from the Switch

TFTP allows you to transfer files between a TFTP server and the following switch storage areas: local file system, internal memory card, compact flash card, and USB 2.0 storage device.

- Download a file from a TFTP server to the switch, using the `tftp` or `tftp get` commands.

  `tftp [ ip-address | host-name ] { -v vr_name } [ -g ] [ { -l local-file | } { -r remote-file } | { -r remote-file } { -l local-file } ]`

  `tftp get [ ip-address | host-name] { vr vr_name } remote-file {local-file} {force-overwrite}`

  Note
  By default, if you transfer a file with a name that already exists on the system, the switch prompts you to overwrite the existing file. For more information, see the `tftp get` command.

- Download a file from a universal resource locator (URL), using the following command:

  `download [url url {vr vrname} | image [active | inactive] [[hostname | ipaddress] filename {{vr} vrname} {block-size block_size}] {partition} {install {reboot}}`

  The `download url` command supports the following file types: `cfg`, `lic`, `lst`, `pol`, `py`, `voss`, `xmod`, `xos`, `xsf`, and `xtr`.

  The `download image` commands supports the following file types: `voss`, `xmod`, `xos`, and `xtr`.

- To upload a file from the switch to a TFTP server, use the `tftp` or `tftp put` commands:

  `tftp [ ip-address | host-name ] { -v vr_name } [ -p ] [ { -l local-file | } { -r remote-file } | { -r remote-file } { -l local-file } ]`

  `tftp put [ ip-address | host-name] {vr vr_name} local-file { remote-file}`

For detailed information about downloading software image files, BootROM files, and switch configurations, see Software Upgrade and Boot Options on page 1697. For more information about configuring core dump files and managing the core dump files stored on your switch, see Understanding Core Dump Messages on page 1710.

Deleting Files from the Switch

You can delete a configuration, policy, or if configured, core dump file from your system by running the `rm file_name` command.

When you delete a configuration or policy file from the system, make sure you specify the appropriate file extension. For example, when you want to delete a policy file, specify the filename and .pol. After you delete a file, it is unavailable to the system.

When you delete a file from the switch, a message similar to the following appears:

```
Remove testpolicy.pol from switch? (y/n)
```

Enter `y` to remove the file from your system. Enter `n` to cancel the process and keep the file on your system.

If you attempt to delete an active configuration file (the configuration currently selected to boot the switch), the switch displays an error similar to the following:

```
Error: Cannot remove current selected active configuration.
```

For more information about configuring core dump files and managing the core dump files stored on your switch, see Understanding Core Dump Messages on page 1710.

Managing the Configuration File

The configuration is the customized set of parameters that you have selected to run on the switch. Table 15 describes some of the key areas of configuration file management in ExtremeXOS.

Table 15: Configuration File Management

Task	Behavior
Configuration file database	ExtremeXOS supports saving a configuration file into any named file and supports more than two saved configurations. For example, you can download a configuration file from a network TFTP server and save that file as primary, secondary, or with a user-defined name. You also select where to save the configuration: primary or secondary partition, or another space.
Downloading configuration files	ExtremeXOS uses the `tftp` and `tftp get` commands to download configuration files from the network TFTP server to the switch. For more information about downloading configuration files, see Using TFTP to Download the Configuration on page 1720.
Uploading configuration files	ExtremeXOS uses the `tftp` and `tftp put` commands to upload configuration files from the switch to the network TFTP server. For more information about uploading configuration files, see Using TFTP to Upload the Configuration on page 1719.

Table 15: Configuration File Management (continued)

Task	Behavior
Managing configuration files, including listing, copying, deleting, and renaming	The following commands allow you to manage configuration files: • `ls`: Lists all of the configuration files in the system. • `cp`: Makes a copy of an existing configuration file in the system. • `rm`: Removes/deletes an existing configuration file from the system. • `mv`: Renames an existing configuration file.
Configuration file types	• XML-formatted configuration file • ASCII-formatted configuration file
XML-formatted configuration file	ExtremeXOS configuration files are saved in Extensible Markup Language (XML) format. Use the `show configuration` command to view on the CLI your currently running switch configuration.
ASCII-formatted configuration file	You can upload your current configuration in ASCII format to a network TFTP server. The uploaded ASCII file retains the CLI format. To view your configuration in ASCII format, save the configuration with the .xsf file extension (known as the XOS CLI script file). This saves the XML-based configuration in an ASCII format readable by a text editor. ExtremeXOS uses the `upload configuration` command to upload the ASCII-formatted configuration file from the switch to the network TFTP server. ExtremeXOS uses the `tftp` and `tftp get` commands to download configuration files from the network TFTP server to the switch. For more information about ASCII-formatted configuration files, see Uploading ASCII-Formatted Configuration Files on page 1715.
XML configuration mode	Indicated by (xml) at the front of the switch prompt. Do not use; instead, run `disable cli xml-mode` to disable this mode.
Displaying configuration files	You can also see a complete list of configuration files by entering the `ls` command followed by the **[Tab]** key.

For more information about saving, uploading, and downloading configuration files, see Save the Configuration on page 1718.

Managing ExtremeXOS Processes

ExtremeXOS consists of a number of cooperating processes running on the switch. With process control, under certain conditions, you can stop and start processes, restart failed processes, examine information about the processes, and update the software for a specific process or set of processes.

Displaying Process Information

To display process information, use the following command:

```
show process {name} {detail} {description} {slot slotid}
```

Stopping Processes

If recommended by Extreme Networks Technical Support personnel, you can stop a running process. You can also use a single command to stop and restart a running process during a software upgrade on the switch.

By using the single command, there is less process disruption and it takes less time to stop and restart the process.

- To stop a running process, use the following command:

  ```
  terminate process name [forceful | graceful]
  ```

  In a SummitStack:

  ```
  terminate process name [forceful | graceful] {slot slot}
  ```

  Where the following is true:

name	Specifies the name of the process to terminate. You can terminate the following processes: bgp, eaps, exsshd, isis, lldp, netLogin, netTools, ntp, ospf, ospfv3, snmpMaster, telnetd, thttpd, tftpd, vrrp, and xmld.
forceful	Specifies a forceful termination.
graceful	Specifies a graceful termination.

Note
Do not terminate a process that was installed since the last reboot unless you have saved your configuration. If you have installed a software module and you terminate the newly installed process without saving your configuration, your module may not be loaded when you attempt to restart the process with the `start process` command.

- To preserve a process's configuration during a terminate and (re)start cycle, save your switch configuration before terminating the process. Do not save the configuration or change the configuration during the process terminate and re(start) cycle. If you save the configuration after terminating a process, and before the process (re)starts, the configuration for that process is lost.
- To stop and restart a process during a software upgrade, use the following command:

  ```
  restart process [class cname | name {msm slot}]
  ```

Where the following is true:

cname	Specifies the name of the process to restart. With this parameter, you can terminate and restart all instances of the process associated with a specific routing protocol on all VRs.You can restart the *OSPF (Open Shortest Path First)* routing protocol and associated processes.
name	Specifies the name of the process to terminate and restart. You can use this command with the following processes: bgp, eaps, exsshd, isis, lldp, netLogin, netTools, ntp, ospf, ospfv3, snmpMaster, telnetd, thttpd, tftpd, vrrp, and xmld.
slot	On a SummitStack, specifies the node's slot number. The number is a value from 1 to 8.

For the config recovery feature to work correctly, the netTools process can't be restarted when config recovery is enabled.

Starting Processes

- To start a process, use the following command:

  `start process` *name* {**msm** *slot*}

  In a SummitStack:

  `start process` *name* {**slot** *slot*}

  Where the following is true:

name	Specifies the name of the process to start. You can start the following processes: bgp, eaps exsshd, isis, lldp, netLogin, netTools, ospf, snmpMaster, telnetd, thttpd, tftpd, vrrp, xmld
slot	On a SummitStack, specifies the node's slot number. The number is a value from 1 to 8.

```
Error: Process telnetd already exists!
```

Note
After you stop a process, do not change the configuration on the switch until you start the process again. A new process loads the configuration that was saved prior to stopping the process. Changes made between a process termination and a process start are lost, and error messages can result when you start the new process.

As described in the section, Stopping Processes on page 141, you can use a single command, rather than multiple commands, to stop and restart a running process.

- Stop and restart a process during a software upgrade.

  `restart process [`**class** *cname* | *name* {**msm** *slot*}]

  In a SummitStack:

  `restart process [`**class** *cname* | *name* {**slot** *slot*}]

  For more detailed information, see Stopping Processes on page 141.

Creating Processes

The following commands allow you to add a process. The process can be a C executable compiled using the C-based SDK or a Python module. You upload it to /usr/local/cfg using the normal mechanisms (for example, TFTP).

To create a C-executable process, use the command:

```
create process name executable exe {start [auto | on-demand]} {node
node} {vr vr-name} {description description} {arg1 {arg2 { arg3 { arg4
{ arg5 { arg6 { arg7 { arg8 { arg9 }}}}}}}}}
```

To create a Python module, use the command:

```
create process name python-module python-module {start [auto | on-
demand]} {node node} {vr vr-name} {description description} {arg1 {arg2
{arg3 {arg4 {arg5 {arg6 {arg7 {arg8 {arg9}}}}}}}}}
```

Understanding Process Control Groups

Processes running in ExtremeXOS are grouped into “Vital” (ExtremeXOS) and “Other” (non-ExtremeXOS) process control groups. The "Vital" (ExtremeXOS) group includes all the ExtremeXOS infrastructure, network services, protocols, configuration and network management processes. The “Other” (non-ExtremeXOS) groups include all the processes created through CLI, and scripts executed in ExtremeXOS shell. Both these groups are capped with memory and CPU limits as a percentage of the total available memory and CPU resources in the system. Both groups get a guaranteed share of memory and CPU access up to that limit.

A user-created process automatically is classified as being in the "Other" group. The "Vital" group has, by default, 95% of the system memory and 90% of the system CPU resources. In contrast, the “Other” group has a default memory limit of 5% and a default CPU limit of 10%, both configurable to a certain extent through CLI. This provides a level of segregation where processes in one group cannot significantly impact processes running in another group.

To manage the CPU and memory usage for the "Other" (non-ExtremeXOS) group, use the following commands:

Note
Increasing the memory or CPU limit of the "Other" group impacts the total available memory or CPU resources of the "EXOS" group. These commands need to be used with careful consideration of the resource requirements in real-time.

- `configure process group other cpu-limit cpu_limit`
- `configure process group other memory-limit memory_limit`

When these commands are issued, the CPU and memory limits for the "EXOS" (ExtremeXOS) group are changed as well.

To clear the user-configured memory and CPU limits and restore default settings, use the following command:

```
unconfigure process group
```

To clear the memory- and CPU-related statistics of "Vital" and/or “Other” groups, use the following command:

```
clear process group statistics {vital | other}
```

To show the configured settings and statistics for the process control groups, use the following command:

```
show process group {[vital | other]}
```

Understanding Memory Protection

ExtremeXOS provides memory management capabilities. Each process in ExtremeXOS runs in a protected memory space. This infrastructure prevents one process from overwriting or corrupting the memory space of another process. For example, if one process experiences a loop condition, is under some type of attack, or is experiencing some type of problem, that process cannot take over or overwrite another processes' memory space.

Memory protection increases the robustness of the system. By isolating and having separate memory space for each individual process, you can more easily identify the process or processes that experience a problem.

To display the current system memory and that of the specified process, use the following command: `show memory process` *`name`* {**`slot`** *`slotid`*}

Where the following is true:

- *`name`*: Specifies the name of the process.
- slot: On a SummitStack, specifies the slot number of the target node. The number is a value from 1 to 8.

Reading the Output

The `show memory process` command displays the following information in a tabular format:

- System memory information (both total and free).
- Current memory used by the individual processes.

The current memory statistics for the individual process also includes the following:

- For SummitStacks, the slot number.
- The name of the process.

Viewing System Memory

You can also use the `show memory {slot slotid}` command to view the system memory and the memory used by the individual processes, even for all processes on all MSMs/MMs installed in modular switches. The slot parameter is available only on modular switches and SummitStack.

In general, the *free* memory count for a switch decreases when one or more running processes experiences an increase in memory usage. If you have not made any system configuration changes, and you observe a continued decrease in free memory, this might indicate a memory leak.

The information from these commands may be useful for your technical support representative if you experience a problem.

The following is sample truncated output from an ExtremeSwitching switch:

```
CPU Utilization Statistics - Monitored every 25 seconds
-------------------------------------------------------------------------
Process         5   10   30    1     5    30   1     Max       Total
              secs secs secs min  mins  mins hour          User/System
              util util util util util util util    util    CPU Usage
              (%)  (%)  (%)  (%)   (%)  (%)  (%)     (%)       (secs)
System        n/a  n/a  0.0  0.9   0.1  0.2  0.5                34.6
-------------------------------------------------------------------------

aaa             n/a  n/a  0.0  0.0  0.0  0.0  0.0  1.8  1.72      0.78

acl             n/a  n/a  0.0  0.0  0.0  0.0  0.0  0.0  0.40      0.24

bgp             n/a  n/a  0.0  0.0  0.0  0.0  0.0  12.6 11.18     2.21

cfgmgr          n/a  n/a  0.0  0.0  0.0  0.0  0.8  39.8 4743.92   3575.79

cli             n/a  n/a  0.0  0.0  0.0  0.0  0.0  0.0  0.59      0.42

devmgr          n/a  n/a  0.0  0.0  0.0  0.0  0.0  19.5 74.44     24.52

dirser          n/a  n/a  0.0  0.0  0.0  0.0  0.0  0.0  0.0       0.0

dosprotect      n/a  n/a  0.0  0.0  0.0  0.0  0.0  0.0  0.8       0.12

eaps            n/a  n/a  0.0  0.0  0.0  0.0  0.1  5.5  36.40     15.41

edp             n/a  n/a  0.0  0.0  0.0  0.0  0.0  11.1 10.92     3.97

elrp            n/a  n/a  0.0  0.0  0.0  0.0  0.0  0.0  0.49      0.44

ems             n/a  n/a  0.0  0.0  0.0  0.0  0.0  0.0  1.19      1.29

epm             n/a  n/a  0.0  0.0  0.0  0.0  0.0  30.7 48.74     32.93

esrp            n/a  n/a  0.0  0.0  0.0  0.0  0.0  2.7  0.82      0.45

etmon           n/a  n/a  0.0  0.0  0.0  0.0  0.5  30.5 4865.78   873.87

...
```

Configuring Stacked Switches

Introduction to Stacking on page 146
Preparing to Configure a Stack on page 159
Configuring a Stack on page 164
Auto-Discovery for Universal Hardware on page 181
Managing an Operational Stack on page 182
Changing the Stack Configuration on page 198
Troubleshooting a Stack on page 211

A stack consists of a group of up to eight switches that are connected to form a ring. The stack offers the combined port capacity of the individual switches. But it operates as if it were a single switch, making network administration easier.

Stacking is facilitated by the SummitStack feature – part of the ExtremeXOS Edge license.

This chapter contains information about configuring a stack, maintaining the stack configuration, and troubleshooting.

For descriptions of the supported configurations for stacking, considerations for planning a stack, and instructions for setting up the hardware, see the *Stacking* chapter in the *hardware installation guide* for your switch. We recommend that you read that chapter before installing the switches that will make up the stack.

Introduction to Stacking

When stacking switches, the stack operates as if it were a single switch with a single IP address and a single point of authentication. One switch – called the primary switch – is responsible for running network protocols and managing the stack. The primary runs Switch Engine software and maintains all the software tables for all the switches in the stack.

All switches in the stack, including the primary switch, are called nodes. Figure 4 on page 148 shows four nodes in a stack, connected to each other by SummitStack cables.

All connections between stack ports must be directly between switches. A stacking connection cannot pass through a third device, for example a Virtual Port Extender or an LRM/MACsec Adapter.

Using the SummitStack feature—part of the Switch Engine Edge/Base license—a stack can combine switches from different series, provided that every switch in the stack:

- Runs in the same partition (primary or secondary).
- Runs the same version of Switch Engine.
- Includes support for stacking.

See for information about which switch series can be combined to form a stack.

The following topics introduce you to the basic principles of stacking and provide recommendations for creating stacks.

More information to answer your questions about stacking and help you plan your configuration is available on the Extreme Networks GTAC Knowledge Base.

Build Basic Stacks

A stack can be created in either of two ways:

- In *native stacking*, switches are connected using either designated Ethernet data ports or dedicated stacking connectors.
- In *alternate stacking*, switches are connected using Ethernet data ports that have been configured for stacking. These ports are located either on the switch itself or on option cards installed on either the front or the rear of the switch.

When planning and building your stack, be sure to follow port compatibility and cabling recommendations as described in this chapter.

Slot Numbers in Stacks

A switch stack can be thought of as a virtual chassis. Each switch (node) operates as if it were occupying a slot in a chassis and is controlled by the primary. The high-speed stacking links function like the backplane links of a chassis.

Each switch in the stack is assigned a “slot number” during the initial software configuration of the stack. Starting at the switch with the console connection, numbers are assigned in numerical order following the physical path of the connected stacking cables. For example, if you follow the cabling recommendations presented in Introduction to Stacking on page 146 and configure a vertical stack from the console on the switch at the top of the physical stack, the switches will be assigned slot numbers 1 through 8 from the top down.

A quick way to verify that the cable connections match the software configuration is to check the stack number indicator on each switch. If the slot numbers do not line up in the order you arranged the switches, this might indicate that the stacking cable setup differs from what you intended when you configured the software. In this case, reconnect the cables in the correct order and perform the software configuration again.

Primary/Backup Switch Redundancy

When your stack is operational, one switch is the primary switch, responsible for running network protocols and managing the stack.

To provide recovery in case of a break in the stack connections, you can configure redundancy by designating a backup switch to take over as primary if the primary switch fails. When you perform the initial software configuration of the stack, the "easy setup" configuration option automatically configures redundancy, with slot 1 as the primary and slot 2 as the backup. You can also configure additional switches as "primary-capable," meaning they can become a stack primary in case the initial backup switch fails.

SummitStack Topologies

Figure 4 presents a graphical representation of a stack and some of the terms that describe stack conditions.

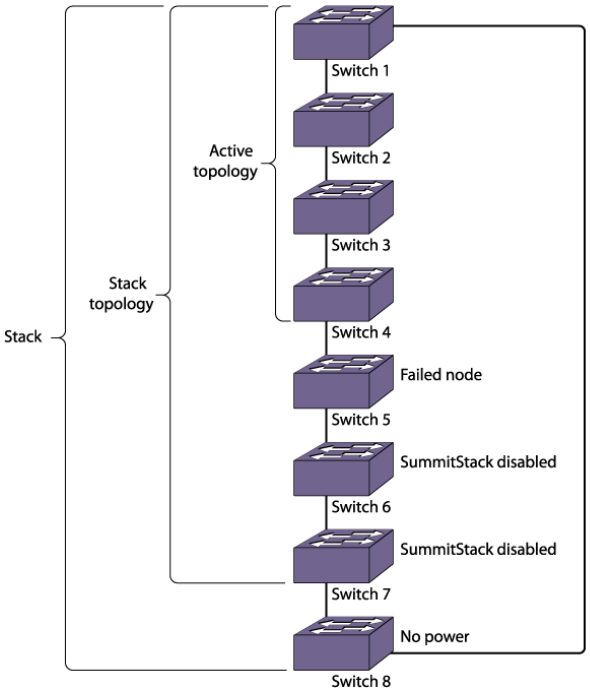


Figure 4: Example of a Stack, Showing the Active Topology and the Stack Topology

A stack is the collection of all switches, or nodes, that are cabled together to form one virtual switch using the Switch Engine SummitStack feature.

The maximum cable length supported between switches depends on the types of switches in your stack, the installed option cards, and the configured stacking ports.

A stack topology is the set of contiguous nodes that are powered up and communicating with each other. In the example shown, Switch 8 is not part of the stack topology because it is not powered up.

An active topology is the set of contiguous nodes that are active. An active node is powered up, is configured for stack operation, and is communicating with the other active nodes.

Switch 5 in the example has failed, stacking is disabled on Switches 6 and 7, and Switch 8 has no power. As a result, the active topology includes Switches 1 through 4 only.

For more information about SummitStack terminology, see SummitStack Terms on page 156.

Ring Topology: Recommended for Stacking

SummitStack nodes should be connected to each other in a ring topology. In a ring topology, one link is used to connect to a node and the other link is used to connect to another node. The result forms a physical ring connection. This topology is highly recommended for normal operation.

Figure 5 represents a maximal ring topology of eight active nodes.

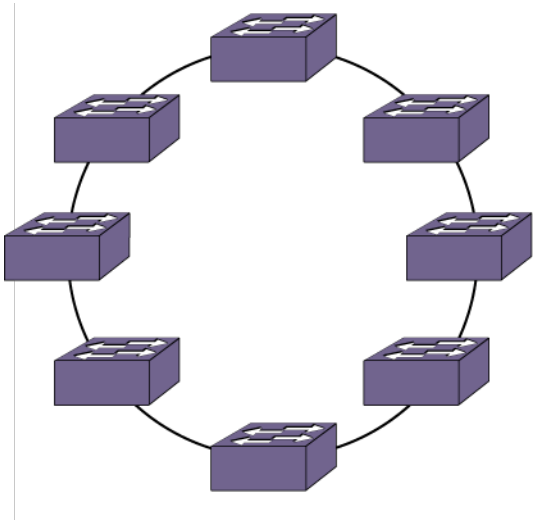


Figure 5: Graphical Representation of a Ring Topology

Each switch in a rack is connected to the switch above it and the switch below it. To complete a ring, a longer cable might be needed to connect the top switch to the bottom switch.

Note that, while a physical ring connection may be present, a ring active topology exists only when all nodes in the stack are active.

Daisy Chain Topology: Not Recommended for Stacking

Stackable switches can be connected in a daisy-chain topology. This is a ring topology with one of the links disconnected, inoperative, or disabled. A daisy chain can be created when a link fails or a node reboots in a ring topology, but the daisy chain topology is not recommended for normal operation.

We strongly recommend that your stack nodes be connected in a ring topology, not a daisy-chain topology, for normal operation.

Nodes in an active topology can operate in a daisy-chain configuration even when there is physically a ring connection in the stack.

You might need to use a daisy chain topology while adding a new node, removing a node, or joining two stacks.

If you are using a daisy chain topology, the possibility of a dual master condition increases. Before you create a daisy chain topology, read Managing a Dual Primary Situation on page 213.

Use Ethernet Ports for Stacking (SummitStack-V Feature)

On some Extreme Networks switches, you can reconfigure Ethernet data ports to operate as stacking ports.

This feature, known as *SummitStack-V* or *alternate stacking*, means that you can use less expensive cables to connect the switches in a stack. Because copper and fiber Ethernet ports support longer cable distances, you can also extend the physical distance between stack nodes – connecting, for example, switches on different floors in a building or in different buildings on a campus.

The SummitStack-V feature means that you can stack switches that have no dedicated (or *native*) stacking ports but that do have at least two Ethernet ports. The ports can be configured to support either data communications or stacking. When configured to support stacking, they are called alternate stacking ports to distinguish them from the native stacking ports.

A single stack can use both native stacking ports and alternate stacking ports. On one switch, for example, you can use a native stacking port to connect to a switch in the same rack, and you can use an alternate stacking port to connect to a switch on a different floor.

Note
When you connect distant nodes using alternate stacking ports, be sure to run the cables over physically different pathways to reduce the likelihood of a cut affecting multiple links.

On each switch model, only specific data ports can be used as alternate stacking ports. The alternate stacking ports can be found on either on the front panel of the switch or on installed versatile interface modules at either the front or rear of the switch. Switch models that do not have native stacking ports can still use alternate stack ports for stacking.

Alternate stacking ports on different switches must be directly connected, with no intervening switch connections. This is because alternate stacking ports use the proprietary Switch Engine protocol for stacking, not the standard Ethernet protocol.

Table 16 lists the data ports that can be used as native and alternate stacking ports for each switch model.

When the stacking-support option is enabled (with the `enable stacking-support` command), data communication stops on the physical data ports that are designated for alternate stacking. Then, when stacking is enabled (with the `enable stacking`

command), those ports – listed in the Alternate Stacking Ports column of Table 16 – operate as stacking ports.

Table 16: Native and Alternate Stacking Ports

Switch Model	Type or Location of Native Stacking Ports	Alternate Stacking Ports	Location of Alternate Stacking Ports
4120-24MW 4120-48MW	Front panel: U1, U2	None	Not applicable
4220-12T 4220-12P 4220-4MW-8P 4220-24T 4220-24P 4220-48T 4220-48P 4220-8MW-40P 4220-8X	Front panel: U1, U2	None	Not applicable
5120-24X-4Y 5120-24XT-4Y	Front panel: U1, U2	None	Not applicable
5120-44X-4Y-2C	Front panel: U1, U2	49, 50	Front panel
5320-16P-4XE 5320-16P-2MXT-2X 5320-16P-4XE-DC 5320-24T-8XE 5320-24P-8XE 5320-48T-8XE 5320-48T-8XE 5320-24T-4X-XT 5320- 24T-24S-4XE-XT	Front panel: U1, U2	None	Not applicable
5520-24T 5520-24W 5520-24X 5520-24T-ACDC-BASE 5520-24X-ACDC-BASE	Front panel: U1, U2	35, 36	5520-VIM-4X
5520-48T 5520-48W 5520-12MW-36W 5520-48SE 5520-48T-ACDC-BASE 5520-48SE-ACDC-BASE	Front panel: U1, U2	59, 60	5520-VIM-4X

Table 16: Native and Alternate Stacking Ports (continued)

Switch Model	Type or Location of Native Stacking Ports	Alternate Stacking Ports	Location of Alternate Stacking Ports
5720-24MW 5720-24MXW 5720-48MW 5720-48MXW	Front panel: U1, U2	None	Not applicable
7520-48Y-8C 7520-48YE-8CE	Front panel: 55, 56	None	Not applicable
7520-48XT-6C	Front panel: 53, 54	None	Not applicable
7720-32C	Front panel: 31, 32	None	Not applicable

Note
Ports designated as U1 and U2 are Universal ports. These ports can be configured as either stacking ports or Ethernet data ports. When stacking support is disabled, the Universal ports will provide normal data communications.

Available Stacking Methods

Each switch model can use various methods of stacking.

Table 17 shows the switch models that can participate in each stacking method.

Table 17: SummitStacking by Stacking Method

Stacking Method	Speed per Link (HDX)	Cable Type and Lengths	Switch Models
SummitStack-V	10 Gbps	0.5 m - 40 km SFP+ (with SR, LR, and ER)	5520 (using 5520-VIM-4X) **Note:** The 5520 switches can stack with 5320 switches or 5420 switches by installing a 5520-VIM-4X into the 5520 switches.
SummitStack-V40	10 Gbps	0.5 m - 40 km SFP+ (with SR, LR, and ER)	4220 (ports U1, U2) **Note:** The 4220 switches can only stack with themselves. 5320 **Note:** The 5320 switches can stack with themselves, the 5420 or 5520 switches using 5520-VIM-4X ports. 5320-24T-4X-XT and 5320-24T-24S-4XE-XT (ports U1, U2) **Note:** 5320-24T-4X-XT and 5320-24T-24S-4XE-XT switches can only stack with themselves. **Note:** 5320-16P-2MXT-2X can stack with other 5320 switches except 5320-24T-4X-XT and 5320-24T-24S-4XE-XT switches. 5420 (ports U1, U2) **Note:** The 5420 switches can stack with themselves, 5320 switches, or 5520 switches using 5520-VIM-4X ports.
SummitStack-V80	20 Gbps	0.5 m - 5 m SFP-DD	5420 (ports U1, U2) **Note:** The 5420 switches can stack with themselves and 5520 (ports U1 and U2) when configured for SummitStack-V80.
SummitStack-V100	25 Gbps	0.5 m - 40 m SFP28 only (with SR, LR, and ER)	5120 (ports U1, U2) **Note:** The 5120 switches can only stack with themselves.

Table 17: SummitStacking by Stacking Method (continued)

Stacking Method	Speed per Link (HDX)	Cable Type and Lengths	Switch Models
SummitStack-V160	40 Gbps	0.5 m - 40 kM QSFP+ only	5520 (ports U1, U2) **Note:** If you use SummitStack-V160, 5520 switches can stack only with themselves. To stack with 5320 and 5420 switches, the 5520 can use either SummitStack-V80 on ports U1 and U2 or SummitStack-V on the alternate stack ports (Using a 5520-VIM-4X).
SummitStack-V200	50 Gbps	0.5 m - 5 m QSFP28 only	4120 (ports U1, U2) **Note:** The 4120 switches can only stack with themselves. 5120-44X-4Y-2C (ports 49, 50) **Note:** The 5120 switches can only stack with themselves. 5520 (ports U1, U2) **Note:** If you use SummitStack-V200, 5520 switches can stack only with themselves or 5720 switches.

Table 17: SummitStacking by Stacking Method (continued)

Stacking Method	Speed per Link (HDX)	Cable Type and Lengths	Switch Models
SummitStack-V400	106 Gbps	0.5 m - 5 m QSFP28 only	5720 (ports U1, U2), Extreme 7520 (ports 53/54 or 55/56), or 7720 (ports 31, 32) **Note:** If you use SummitStack-V400, 5720 switches can only stack with themselves. **Note:** No VIM ports can be configured as stack ports for 5720 switches. **Note:** If you use SummitStack-V400, 7520 and 7720 switches can only stack with themselves.
SummitStack-V400 Alternative Configuration	100 Gbps	V400 Alternative Configuration is required when using specific fiber cables. This mode sets the stack ports to 100G, enables pre-emphasis, and FEC (clause_91). • QSFP28 SR4 • QSFP28 LR4 • QSFP28 CWDM4 • QSFP28 PSM4 • QxQ AOC cable - 5m • QxQ AOC cable - 7m • QxQ AOC cable - 10m • QxQ AOC cable - 20m	5720 (ports U1, U2), Extreme 7520 (ports 53/54 or 55/56), or 7720 (ports 31, 32) **Note:** If you use SummitStack-V400 Alternative Configuration, 5720 switches can only stack with themselves. **Note:** No VIM ports can be configured as stack ports for 5720 switches. **Note:** If you use SummitStack-V400 Alternative Configuration, 7520 and 7720 switches can only stack with themselves.

For more details about the stacking methods that are available for each switch series, see the hardware installation guide for your switch.

Note
All switches in the stack must run the same version of Switch Engine.

SummitStack Terms

Table 18 describes the terms used for the SummitStack feature. These terms are listed in the recommended reading sequence.

Table 18: List of Stacking Terms

Term	Description
Stackable switch	An Extreme Networks switch that provides two stacking ports and can participate in a stack.
Stacking port	A physical interface of a stackable switch that is used to allow the connection of a stacking link. Stacking ports are point-to-point links that are dedicated for the purpose of forming a stack.
Native stacking	A stacking configuration in which stack members are connected using either designated Ethernet data ports or dedicated stacking connectors.
Alternate stacking	A stacking configuration in which stack members are connected using 10-Gbps Ethernet data ports that have been configured for stacking. These ports are located either on the switch itself or on option cards installed on the rear of the switch.
Stacking link	A cable that connects a stacking port of one stackable switch to a stacking port of another stackable switch, plus the stacking ports themselves.
Node	A switch that runs the Switch Engine operating system and is part of a stack. Synonymous with *stackable switch*.
Stack	A set of stackable switches and their connected stacking links made with the intentions that: (1) all switches are reachable through their common connections; (2) a single stackable switch can manage the entire stack; and (3) configurable entities such as VLANs and link trunk groups can have members on multiple stackable switches. A stack consists of all connected nodes regardless of the state of the nodes.
Stack topology	A contiguously connected set of nodes in a stack that are currently communicating with one another. All nodes that appear in the `show stacking` command display are present in the stack topology.
Stack path	A data path that is formed over the stacking links for the purpose of determining the set of nodes that are present in the stack topology and their locations in the stack. Every node is always present in a stack path whether or not stacking is enabled on the node.
Control path	A data path that is formed over the stacking links that is dedicated to carrying control traffic, such as commands to program hardware or software image data for software upgrade. A node must join the control path to fully operate in the stack. A node that is disabled for stacking does not join the control path, but does communicate over the stack path.

Table 18: List of Stacking Terms (continued)

Term	Description
Active node	A node that has joined the control path. The active node can forward the control path messages or can process them. It can also forward data traffic. Only an active node can appear as a card inserted into a slot when the `show slot {slot {detail} | detail }` command is executed on the primary node of the stack.
Active topology	A contiguous set of active nodes in a stack topology plus the set of stacking links that connect them. When an active topology consists of more than one node, each node in the active topology is directly and physically connected to at least one other node in the active topology. Thus, the active topology is a set of physically contiguous active nodes within a stack topology.
Candidate node	A node that is a potential member of an active topology, or an active node that is already a member of an active topology. A candidate node may or may not be an active mode – that is, it may or may not have joined the control path.
Node role	The role that each active node plays in the stack – either primary, backup, or standby.
Primary node	The node that is elected as the primary node in the stack. The primary node runs all of the configured control protocols such as OSPF (Open Shortest Path First), RIP (Routing Information Protocol), Spanning Tree, and EAPS (Extreme Automatic Protection Switching). The primary node controls all of its own data ports as well as all data ports on the backup and standby nodes. To accomplish this, the primary node issues specific programming commands over the control path to the backup and standby nodes.
Backup node	The node assigned to take over the role of primary if the primary node fails. The primary node keeps the backup node's databases synchronized with its own databases in preparation for such an event. If and when the primary node fails, the backup node becomes the primary node and begins operating with the databases it has previously received. In this way, all other nodes in the stack can continue operating.
Standby node	A node that is prepared to become a backup node in the event that the backup node becomes the primary node. When a backup node becomes a primary node, the new primary node synchronizes all of its databases to the new backup node. When a node operates in a standby role, most databases are not synchronized – except those few that directly relate to hardware programming.

Table 18: List of Stacking Terms (continued)

Term	Description
Acquired node	A standby or backup node that is acquired by a primary node. This means that the primary node has used its databases to program the hardware of the standby or backup node. The standby or backup node has acted as a hardware programming proxy, accepting the instructions of the primary node to do so. An acquired backup node maintains the databases needed to reflect why the hardware is programmed as it is. However, a standby node does not. An acquired node can be re-acquired (without a reboot) by the backup node only when the backup node becomes the primary node, and only when both the backup and standby nodes were already acquired by the same primary node at the time of its failure.
Data ports	The set of ports on a stackable switch that are available for connection to your data networks. Such ports can be members of a user-configured VLAN or trunk group. They can be used for Layer 2 and 3 forwarding of user data traffic, for mirroring, or other features you can configure. Data ports are different from stacking ports.
Failover	The process of changing the backup node to the primary node when the original primary node has failed. When a primary node fails, if a backup node is present, and if that node has completed its initial synchronization with the primary node, then the backup node assumes the role of primary node. The standby nodes continue their operation and their data ports do not fail.
Hitless failover	A failover in which all data ports in the stack, except those of the failing primary node, continue normal operation when the primary node fails.
Node address	The unique MAC address that is factory-assigned to each node.
Node role election	The process that determines the role for each node. The election takes place during initial stack startup and elects one primary node and one backup node. An election also takes place after a primary node failover, when a new backup node is elected from the remaining standby nodes.
Node role election priority	A priority assigned to each node, to be used in node role election. The node with the highest node role election priority during a role election becomes the primary node. The node with the second highest node role election priority becomes the backup.
Operational node	A node that has achieved operational state as a card in a slot. The operational state can be displayed using the `show slot {slot {detail} | detail }`command.
System uptime	The amount of time that has passed since the last node role election. You can display the system uptime by entering the `show switch {detail }` command on the primary node.

Table 18: List of Stacking Terms (continued)

Term	Description
Stack segment	A collection of nodes that form a stack topology. The term is useful when a stack is severed. Each severed portion of the stack is referred to as a stack segment.
Stack state	A state assigned by the stack to a node. You can display the stack state by entering the `show stacking` command.
Easy Setup	A procedure that automatically configures the essential stacking parameters on every node for initial stack deployment, and then automatically reboots the stack to put the parameters into effect. The choice to run Easy Setup is offered when you run the `enable stacking {node-address` *`node-address`*`}` command and the essential stacking parameters are unconfigured or inconsistent. It can also be invoked directly by running the `configure stacking easy-setup` command.

Preparing to Configure a Stack

The following topics contain background information to help you configure your stack so that it functions as effectively as possible:

- Stacking Node Roles, Redundancy, and Failover on page 159
- Stack Configuration Parameters, Configuration Files, and Port Numbering on page 160
- QoS in Stacking on page 161
- Stacking Link Overcommitment on page 163
- Log Messages from Stack Nodes on page 163

Stacking Node Roles, Redundancy, and Failover

ExtremeXOS supports control plane redundancy and hitless failover.

A stack supports control plane redundancy and hitless failover. Hitless failover is supported to the extent that the failing master node and all of its ports are operationally lost, including the loss of supplied power on any *PoE (Power over Ethernet)* ports that the node provided, but all other nodes and their provided ports continue to operate. After the failover, the backup node becomes the master node.

At failover time, a new backup node is selected from the remaining standby nodes that are configured to be master capable. All operational databases are then synchronized from the new master node to the new backup node. Another hitless failover is possible only after the initial synchronization to the new backup node has completed. This can be seen using the `show switch {detail}` command on the master node and noting that the new backup node is In Sync.

When a backup node transitions to the master node role, it activates the Management IP interface that is common to the whole stack. If you have correctly configured an alternate management IP address, the IP address remains reachable.

When a standby node is acquired by a master node, the standby node learns the identity of its backup node. The master node synchronizes a minimal subset of its databases with the standby nodes.

When a standby node loses contact with both its acquiring master and backup nodes, it reboots.

A master node that detects the loss of an acquired standby node indicates that the slot the standby node occupied is now empty and flushes its dynamic databases of all information previously learned about the lost standby node.

A backup node restarts if the backup node has not completed its initial synchronization with the master node before the master node is lost. When a backup node transitions to the master node role and detects that the master node has not already synchronized a minimal subset of its databases with a standby node, the standby node is restarted.

Reboot or Failure of a Non-Master Node

If a backup node fails, a standby node configured as master-capable is elected as the new backup. This new backup node is then synchronized to the databases of the master node.

For all non-master nodes, a node that reboots or is power-cycled loses all of its connections to all networks for the duration of the reboot cycle. Any *PoE* ports that were providing power prior to the event do not supply power.

When a non-master node fails, the master node marks the related slot as Empty. All other nodes exclude the failed node from the control path and any customer-configured VLANs, trunk group ports, mirroring ports, and so forth.

Stack Configuration Parameters, Configuration Files, and Port Numbering

The stacking configurations are stored in the NVRAM of each node. Some of these configurations take effect only during the next node restart.

Table 19: Stacking Configuration Items, Time of Effect, and Default Value

Configuration Item	Takes Effect	Default Value
Stacking Mode	at boot time	Disabled
Slot Number	at boot time	1
Master-Capable	at boot time	Yes
License Restriction	at boot time	Not configured
Priority	at the next master election	Automatic
Alternate IP Address	immediately	Not configured
Stack MAC	at boot time	Not configured

Stacking parameters, such as mode, slot number, etc., can be configured from a single unit in the stack topology. You can change the stacking-specific configuration even when a node is not in stacking mode, but is connected to the stack. The target node

for the configuration must be powered on and running a version of ExtremeXOS that supports stacking. Further, the node need not be in stacking mode and can be in any node role.

Most ExtremeXOS configuration parameters are not stored in NVRAM, but are instead stored in a configuration file. Configurations stored in NVRAM are those that are needed when the configuration file is not available. The configuration file chosen for the stack is the one selected on the master node that is first elected after a stack restart.

The data (non-stacking) port numbers, in the existing configuration files (which were created when not in stacking mode), are simple integer quantities. On a stack, the data port numbers are expressed as slot:port; where the slot is an integer representing the slot and port is an integer representing the port. For example, 1:2. The configuration file contains an indication that it was created on a stackable switch in stacking mode. The indication is the stacking platform ID. Thus, when in stacking mode, the ports are referenced in the configuration file with the slot:port notation and when not in stacking mode, the ports are referenced as simple integers.

When a standalone switch is configured for stacking, after reboot the non-slotted configuration is converted to a single slot (slot = 1) configuration (all port numbers and port ranges are prepended with "1:"). The configuration is not automatically saved. You must save the configuration to overwrite the existing `primary.cfg`, `secondary.cfg`, or custom named file.

QoS in Stacking

Each stack uses *QoS (Quality of Service)* on the stacking links to prioritize the following traffic within the stack:

- Stack topology control packets
- ExtremeXOS control packets
- Data packets

For stack performance and reliability, the priority of control packets is elevated over that of data packets.

This is done to prevent control packet loss and avoid the timed retries that can lower performance. It is also done to prevent unneeded stack topology changes that can occur if enough stack topology information packets are lost. For these reasons, the SummitStack feature reserves one QoS profile to provide higher priority to control packets. The following sections describe the differences in QoS while using it in stack.

QoS Profile Restrictions

In stacking mode, CoS level 6 (hardware queue 6) is reserved for stacking, so you cannot create quality profile QP7.

Because QP7 cannot be created, you cannot use hardware queue 6 to assign CoS level 6 to a packet. However, you can assign packets received with 802.1p priority 6 to a *QoS*

profile using the technique described in Processing of Packets Received With 802.1p Priority 6 on page 162

Note
This restriction is applicable only when the stackable switch is operating in stacking mode.

QoS Scheduler Operation

In stacking mode, the *QoS* scheduler operation is different for the stacking ports and the data ports.

The scheduler for the data ports operates the same as for standalone ExtremeSwitching switches and is managed with the following command:

```
configure qosscheduler [default |strict-priority | weighted-round-robin
| weighteddeficit- round-robin]
```

The scheduler for the stacking ports is defined by the software when the stack is configured, and it cannot be modified. For all switches, the scheduler is set to strict-priority for the stacking ports, and meters are used to elevate the queue 6 priority above the priority of the other queues. This is the only scheduling method for stack ports.

Processing of Packets Received With 802.1p Priority 6

By default, 802.1p examination is turned on.

Priority 7 is mapped to *QoS* profile QP8, and priorities 6 through 0 are mapped to QoS profile QP1. You can create other QoS profiles and can change this mapping as needed. Since you cannot create QP7 in stacking mode, 802.1p examination always maps packets with priority 6 to other QoS levels. However, you can use an *ACL (Access Control List)* rule entry to set the 802.1p egress value to 6 without affecting the QoS profile assignment as shown in this example:

```
entry VoIPinSummitStack { if { IP-TOS 46; } then { replace-dot1p-value 6; } }
```

Effects on 802.1p Examination

You can turn off 802.1p examination.

When stacking is enabled, the examination remains turned on for priority 6. However, the examination happens at a lower precedence than that of all other traffic groupings.

The mapping you have configured for priority 6 remains in effect, and changes accordingly if you subsequently change the mapping.

When stacking is not enabled, all 802.1p examination is disabled when the feature is turned off.

Effects on DiffServ Examination

When DiffServ examination and 802.1p examination are both turned off, the 802.1p examination for packets arriving at 802.1p priority level 6 remains on at the lowered precedence.

In addition, the examination is adjusted to apply to all packets. The actual priority levels that are used for such packets are the defaults (QP1), or the values last configured using the following command:

```
configure dot1p type dot1p_priority {qosprofile} qosprofile
```

Effects on Port QoS and VLAN QoS

Port *QoS* and *VLAN (Virtual LAN)* QoS have a higher precedence than the 802.1p priority examination performed when the 802.1p examination feature is turned off, and is therefore unaffected.

Stacking Link Overcommitment

The stack is formed by each node supplying a pair of full-duplex, logical stacking ports. Each node can operate on a stack with full duplex throughput up to the limits found in the *hardware installation guide* for your switch.

Even though two links are available, the links might not be fully utilized. For example, suppose there is a ring of eight nodes and the nodes are numbered clockwise from 1 to 8. The stacking port limit in this example is 10 Gbps in each direction for a total stack throughput of 20 Gbps for each port, or 40 Gbps total. Suppose node 1 wants to send 10 Gbps of unicast traffic to each of node 2 and node 3. The shortest path topology forces all traffic from node 1 over the link to node 2. Traffic from node 1 to node 3 passes through node 2. Thus, there is only 10 Gbps link available. However, if node 1 wanted to send 10 Gbps to node 2 and node 8, there would be 20 Gbps available because both links connected to node 1 would be used.

In a ring of eight nodes, between any two nodes, only one link is used. If the devices provide 48 1-Gbps Ethernet ports, the overcommitment ratio between two such nodes is approximately 5:1.

The following command displays stacking link overcommitment information:

```
show ports stack-ports stack_port_list congestion {no-refresh | refresh}
port-number
```

Log Messages from Stack Nodes

Each node can generate log messages through the usual logging mechanism.

On backup and standby nodes, a log target and related filter is automatically installed. The log target is the master node. The filter allows all messages that have a log level of warning, error, or critical to be saved in the log file of the master node.

If the master node changes, the log target is updated on all the remaining nodes. You can also log in to any node in the active topology and see the complete log of the node.

Configuring a Stack

Before configuring a new stack, do the following:

- Ensure that every switch, or node, in the stack is running on the same partition (primary or secondary). To find this information for any node, issue the command `show switch` and look for the `Image Booted` field in the output.
- Ensure that every switch in the stack is running the same version *and patch level* of ExtremeXOS. To find this information for any node, issue the command `show switch` and look for the `Primary Ver` or `Secondary Ver` field in the output.
- If necessary, enable switch license levels or configure license level restrictions for the nodes in the stack.
  - To understand how licensing works in stacked switches, see Managing Licenses on a Stack on page 184.
  - To enable a license on a node, see Software Upgrade and Boot Options on page 1697.
  - To restrict the license level for one or more nodes in a stack, see Restricting a Switch License Level on page 186.
- We recommend that you save the configuration for each switch that will participate in the stack, so that you can reinstate the configuration after the switch is no longer needed in the stack.

Note
New switches are primary-capable by default.

Because stacks can consist of switches of different series and different models, ExtremeXOS does not restrict configuration settings based on the capabilities of any particular node in the stack. Therefore, you are responsible for ensuring that your configuration settings are appropriate for all switches in the stack.

The following switch families come from the factory with stacking-support enabled with stack ports set to a default configuration. The following table describes the default configuration for each switch:

Table 20: Universal Switch Default Stack Port Configuration

Switch	Stack port default configuration
4120	SummitStack V200
4220	SummitStack V40
5320	SummitStack V40
5420	SummitStack V40

Table 20: Universal Switch Default Stack Port Configuration (continued)

Switch	Stack port default configuration
5520	SummitStack V160
5720	SummitStack V400

Important

Some switches contain different types of EXOS images. 5720, 7520, and 7720 require an `onie_x86_64` image type. The 4220 Series, 5320, 5320 extended temperature (XT), 5420, and 5520 require a `summit_arm` image type. The 4120 and 5120 Series require an `rzg2` image type. Before building a mixed stack of these switch families, all switches must have the same release level image loaded onto all nodes.

Once a mixed stack of 5720 and 5520 switch families has been built, a software upgrade is required. Download and install the `onie-summitarm-bundle.lst` image, which combines the `onie_x86_64` image and the `summit_arm` image into one bundle.

Alternatively, you can use the following two step upgrade process:

1. Download the `onie_x86_64` image (and optionally install) onto the Primary node in the stack.
2. Download the `summit_arm` image (and optionally install) onto the Primary node in the stack.

Follow these steps to configure a new stack. Some of the steps include references where you can find additional information.

1. Physically connect the switches (stack nodes) using their stacking ports or alternate stacking ports.

   Instructions for physically setting up the stack are provided in the *hardware installation guide* for your switch.
2. Power on the switches.
3. On each switch, issue the command `enable stacking-support`.

   This command configures the switch so that it is capable of being added to a stack. It is not the same as `enable stacking`, which you will use later to build the stack.
4. Ensure that all of the cabled stacking ports are enabled for stacking:
   a. On each node, issue the command `show ports` *x*`,`*y*, where *x* and *y* are the port numbers.

      In the output, `port state = E` means a port is enabled.
   b. For any ports that are *not* enabled, issue the command `enable ports` *x*`,`*y*, where *x* and *y* are the port numbers.
   c. If you enabled ports on any switches, reboot those switches.

      Save the configuration files.

5. Configure the stack port speed for each stack port on each switch in the stack. Refer to Table 17 on page 153 for the supported speed of each Summit switch model.

   Note
   If the stack will use MPLS, only the 5520 switch can act as primary and backup.

   a. Configure switches that will use alternate stacking ports as described in Use Ethernet Ports for Stacking (SummitStack-V Feature) on page 150.
   b. Reboot the switches whose configurations you changed.
6. Log in, using the console port, to the switch that will be the primary.
7. Issue the command `show stacking stack-ports` to verify that the stacking ports are properly connected.

   All of the ports should show a state of `Operational`.

   In the following example, two of the ports are not in an `Operational` state, which means they are not connected properly.

```
* switch1 # show stacking stack-ports
Stack Topology is a Ring
Slot Port Select Node MAC Address  Port State  Flags Speed
---- ---- ------ ----------------- ----------- ----- -----
*-   1    27     00:04:96:9c:e4:39 Operational C-      10G
*-   2    28     00:04:96:9c:e4:39 Operational CB      10G
 -   1    27     00:04:96:9b:c1:34 Operational CB      10G
 -   2    28     00:04:96:9b:c1:34 Operational C-      10G
 -   1    15     00:04:96:9e:5c:76 Link Down   C-      10G
 -   2    16     00:04:96:9e:5c:76 No-Neighbor C-      10G
 -   1    15     00:04:96:9c:53:b6 Operational C-      10G
 -   2    16     00:04:96:9c:53:b6 Operational C-      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is Blocked
```

   When a port displays a state other than `Operational`, use the following tips to troubleshoot:

   - If the state displays as `Link Down`, check the port's physical connections.
   - Verify that the port is using the same stacking technology, for example SummitStack-V160, as the rest of the ports in the stack.
8. Verify that all nodes in the stack have stacking disabled.

   Note that, even though you enabled stacking-support in step 3 on page 165, it is necessary that stacking be *disabled* before the stack is actually built.

   a. Issue the command `show stacking configuration`.

      For example:

```
* 5420F-48P-4XE # show stacking configuration
Stack MAC in use: <none>
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway          Flags     Lic
------------------ --- --- ---- ------------------ ---------------- --------- ---
*00:04:96:9c:e4:39 1   -   Auto <none>             <none>           -c-----Nn --
 00:04:96:9b:c1:34 1   -   Auto <none>             <none>           -c-----Nn --
 00:04:96:9e:5c:76 1   -   Auto <none>             <none>           -c-----Nn --
 00:04:96:9c:53:b6 1   -   Auto <none>             <none>           -c-----Nn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
```

```
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured
```

b. If a node is enabled for stacking (shown by a capital letter `E` in the Flags column), issue the command `disable stacking node-address` *`mac_address`* for that node.

   Then reboot the switch.

9. From the node that will be the primary, issue the command `enable stacking`.

   The following prompt displays:

```
5420F-48P-4XE.4 # enable stacking
You have not yet configured all required stacking parameters.
Would you like to perform an easy setup for stacking operation? (y/N)
```

10. Enter `y` to proceed with Easy Setup (strongly recommended), or enter `n` to configure the stack manually.

    - When you enter `y` for Easy Setup, you are prompted to confirm your choice:

```
Executing "configure stacking easy-setup" command...
For every node in the 4-node stack, this command will:
- enable stacking
- configure a stack MAC address
- choose and configure a slot number (this node will be assigned to slot 1)
- configure redundancy to minimal (slot 1 will be the master node)
- configure the stacking protocol to enhanced
Upon completion, the stack will automatically be rebooted into the new
configuration.
Warning: If stacking is already configured, this command will alter that
configuration.
Do you wish to proceed? (y/N)
```

      Enter `y` again to confirm. All of the switches reboot automatically and form a stack in which one switch is the primary, one is the backup, and the rest are standby nodes. Easy Setup configures all other required stacking parameters for every switch in the stack.

    - If you forgo Easy Setup by entering `n` in response to the prompt (not recommended), you will need to configure the stack manually. Instead of continuing with step 11, follow the steps in Manually Configuring a Stack on page 168.

11. Verify the configuration, following the instructions in Verifying the Configuration on page 175.
12. Save the ExtremeXOS configuration to every active node in the stack.

    On the primary node, issue the command `save configuration` *`config_name`*, where *`config_name`* is a descriptive name for this configuration.

    The stacking-specific configuration parameters are saved in a file called *`config_name`*`.cfg` to the NVRAM of each node.

The stack is ready to use.

Manually Configuring a Stack

We recommend that you configure your stack using Easy Setup, as described in step 9 on page 167.

However, instead of running Easy Setup, you can configure the stack parameters manually. After performing step 1 on page 165 through step 9 on page 167, perform the following steps as needed:

1. Reboot the switch that will be the master.
2. Optionally, assign slot numbers to all switches in the stack.
   See Configuring Slot Numbers on page 168.
3. Configure node priorities on each slot.
   When the stack boots up, the node priority determines which node will be the master and which node will be the backup. Node priorities can be from 1 to 99, the lowest numbered slot having the highest priority.

   See Configuring the Primary, Backup, and Standby Roles on page 169.
4. Disable master capability for any nodes that you do not want to become master nodes.
   See Configuring Master-Capability on page 197.
5. Assign a MAC address to the stack.
   See Assigning a MAC Address to a Stack on page 172.
6. Optionally, configure a failsafe account for the stack.
   See Failsafe Accounts on page 55.
7. Optionally, set a command prompt for the stack.
   Issue the command `configure snmp sysName stack_name`.

   If you do not define your own command prompt, the default command prompt looks similar to

   `* Slot-6 Stack.9 #`, where:

   - * indicates a changed and unsaved ExtremeXOS configuration
   - 9 is a sequence number indicating the 9th command to be entered since login
   - # indicates that your are logged into the master node (other nodes display the > symbol)
8. When you have performed all desired configuration steps, reboot the stack.
9. Verify the configuration, following the instructions in Verifying the Configuration on page 175.
10. Save the ExtremeXOS configuration to every active node in the stack.
    On the primary node, issue the command `save configuration config_name`, where `config_name` is a descriptive name for this configuration.

    The stacking-specific configuration parameters are saved in a file called `config_name.cfg` to the NVRAM of each node.

Configuring Slot Numbers

When you configure a stack manually, each node in the stack must be assigned a unique slot number. You can assign the slot number only through configuration; the

stack does not dynamically assign a slot number. The available slot numbers are 1 through 8. You can specify a slot number for each node manually, or you can have the system assign the slot numbers using a single command.

Note
Slot numbers take effect only after a restart. If you change a slot number, the unit continues to operate with the slot number it was last restarted with.

- To manually add a slot number to a node, use the following command:

  `configure stacking node-address` *`mac_address`* `slot-number` *`slot_number`*

  where *`mac_address`* is the node's MAC address.
- To configure the system to choose slot numbers for all nodes, enter the command:

  `configure stacking slot-number automatic`

  Automatic slot number assignment is performed in the order of appearance of the nodes in the `show stacking` display. In the case of a ring topology, the first node in the display is the intended master node into which you have logged in.
- To view current slot number assignments, regardless of whether they were assigned manually or automatically, use either the `show stacking` or the `show stacking configuration` command.

  You can find examples of these commands and their output in Verifying the Configuration on page 175.

  Note
  A node that boots in standalone mode does not use a slot number.

Configuring the Primary, Backup, and Standby Roles

Each stack has a primary node, and it might have a backup node and multiple standby nodes. Each switch's role is determined by its priority.

The role of each stack node is determined by, in order of precedence:

- The switch model number.
- The configured priority value (a higher value means the switch has higher priority).
- The slot number in the stack (slot 1 is the primary, slot 2 is the backup).
- The setting of the **`master-capability`** option (see Configuring Master-Capability on page 197).

Some switch models have greater CPU processing capability, more memory, and support additional features – thus making them more suitable for the role of primary node.

To support the additional capabilities in a stack that includes multiple switch models, the most capable switch automatically becomes the primary node. When choosing a switch to act as a backup node, a switch with the same capability as the primary node should be chosen.

The following list shows Universal switch models that can cross stack with each other. Within the list, the relative ranking is shown with the most capable switches placed at the top of the list:

1. 7720 Series (most capable)
2. 7520 Series (most capable)

Note
The 7720 Series and 7520 Series have the same level of capability. Either switch can perform the role of Primary node or Backup node when combined in a mixed stack.

1. 5720 Series
   a. 5720-MXW Series (most capable)
   b. 5720-MW Series
2. 5520 Series

When building a mixed stack of 5720-MXW and 5720-MW switches, choose the **Primary** and **Backup** switches from the same tier—either MXW/MXW or MW/MW. Do not mix tiers for these roles. If you're operating a mixed stack of 5520 and 5720 switches, assign the **Primary** and **Backup** roles to the **5720** switches. The 5520 switches can only function as **Standby** nodes in this configuration.

1. 5520 Series (most capable)
2. 5420 Series
3. 5320 Series

Note
5320-24T-4X-XT and 532024T-24S-4XE-XT switches can only stack with themselves. These switches are designated as Extended Temperature and have "XT" in their model number. These switches can't cross stack with any other series of switches including the non-XT series of 5320 switches.

Important
When stacking standard 5320 switch with 5320-16P-2MXT-2X switches, the **Primary** and **Backup** nodes must be from the same model type:

- Standard 5320 + Standard 5320
- 5320-16P-2MXT-2X + 5320-16P-2MXT-2X
- Do not mix model types for Primary and Backup roles.

Standby nodes may be any combination of models. Only the Primary and Backup nodes should have Primary-capability enabled; standby nodes must not.

Note

In mixed stacks containing both Standard and 5320-16P-2MXT-2X switches, automatic image upgrades between model types are not supported. Similarly, when adding a 5320-16P-2MXT-2X switch to a mixed stack that includes 5520 or 5420 switches, automatic image upgrade for the 5320-16P-2MXT-2X is not supported. If a mismatch is detected, the system will display a warning and skip the upgrade.

To proceed, manually upgrade the image on the affected switch using the `download image` CLI command.

Note

In version 33.3.1 and earlier, the clone script used by `ZTPStack` and `sync slot` did not support downgrades from 33.4.1. To enable downgrade functionality:

- Load both the **ZTPStack XMOD** and **clone XMOD** onto the **Primary switch**.

Once loaded, both `ZTPStack` and `sync slot` will perform the downgrade correctly.

If the stack configuration includes switches that are more capable than others, the stack will try to select the most-capable backup node.

If a switch with reduced capabilities serves as the backup node for a switch with greater capabilities, that switch might not be able to support the stack as a primary node if a failover occurs (for example, the less-capable switch might not have enough processing power or table space to run efficiently as the primary node). If your configuration needs to support automatic failover, we recommend that if a stack contains mixed model numbers, one of the following configurations should be used:

- Identical, most-capable switches available to become the primary and backup nodes.
- The primary-capability option is turned off for all less-capable switches.

When all primary-capable nodes in a stack have the same model number, the node with the highest node role election priority becomes the primary as a result of the first node role election, and the node with the second highest node role election priority becomes the backup node. All other nodes become standby nodes. See Node Election on page 85 for more information.

During subsequent node role elections that occur when a primary node fails, the node priority configuration helps determine the node that becomes the replacement backup node.

Node priority configuration takes effect at the next node role election. A change in node priority configuration does not cause a new election. Once an active topology has elected a primary node, that node retains the primary node role until it fails or loses a dual primary resolution.

You can configure one of the following election priority algorithms:

- Priority algorithm: If any node has a numeric priority value configured. Use the `configure stacking priority` command to configure priority values for the nodes in the stack.
- Automatic algorithm: If all nodes participating in node role election have the automatic priority value configured.

The priority algorithm is selected if any node has a numeric priority value configured. You can specify an integer priority value between 1 and 100. The higher the value, the greater the node role election priority. For example, the command `configure stacking slot 2 priority 10` assigns a priority of 10 to the switch in slot 2.

If any node participating in a role election has a priority value configured, all nodes use the priority algorithm. A node configured with the automatic algorithm uses a priority value of zero (the lowest priority) in the priority algorithm if another node has a priority value configured.

The automatic algorithm is selected if no node participating in a role election has a numeric priority value configured. In automatic mode, the stack determines the highest role election priority based on factors such as available processing power, maintenance level of ExtremeXOS, and so forth.

In both algorithms, if the highest computed node role election priority is shared among multiple nodes, the slot number is used to adjust the node role election priority. A numerically lower slot number results in a higher role election priority than a numerically higher slot number. If you want to use the slot number as the sole determining factor in node role election priority calculation, you should configure every node with the same priority value, rather than using the automatic algorithm.

Note
The automatic priority algorithm may change in future ExtremeXOS releases.

Nodes that are configured as not primary-capable do not participate in node role election. Priority configuration is not relevant on such nodes.

A dual primary resolution does not use the configured node priority in most cases. Instead, it uses the oldest time that a node became a primary in the current active topology.

Assigning a MAC Address to a Stack

Each stack must use a single MAC address. When the master node fails over to the backup node, the backup node must continue to use the same MAC address that the master node was using.

Each stackable switch is assigned a single unique MAC address during production. By default, no stack MAC address is configured. You can choose any node to supply its factory assigned MAC address to form the stack MAC address.

Note
This task is not necessary when you configure the stack using Easy Setup. With Easy Setup, the MAC address is assigned by default.

When you assign a MAC address to a stack, one of the stackable switches is designated as the node whose factory-assigned MAC address is used to form the stack MAC address. Once this is done, all nodes receive and store this formed MAC address in their own NVRAM. Whenever the stack boots up, this MAC address is used, regardless of which node is the master.

When new nodes are added to the stack, the new nodes must be configured with the stack MAC address. The easiest way to do this is to use the `synchronize stacking {node-address node_address | slot slot_number}` command.

Before being stored as the stack MAC address, the chosen node's factory-assigned MAC address is converted to a locally administered MAC address. This prevents duplicate MAC address problems which lead to dual master conditions. The chosen MAC address is put into effect only at node boot time. If the address needs to be changed on a single node, rebooting that node results in usage of the same address stack-wide.

If you do not configure the stack MAC address or it is not the same on all nodes, a warning message appears in the log.

Each node operates with whatever address is available: the configured stack MAC address or the node's factory-assigned MAC address. If a master node fails over to the backup node, and the backup node's address is different than the one the former master node was using, the address is inconsistent with the addresses programmed into the packet forwarding hardware. The MAC address related to the management IP address changes to the one in use by the new master, but no gratuitous ARP requests are sent. In this case, it takes some time for hosts on the management network to flush the related ARP entry.

Note
If the node whose MAC address is chosen was removed from the stack with the intention of using the node elsewhere in the network, and that node is selected to supply the stack MAC in its new stack, the stack MAC of the original stack must be reconfigured to prevent a duplicate MAC address in the network.

To assign a MAC address to a stack, follow these steps.

1. Use the `show stacking configuration` command to display the stack MAC address configuration.

```
Slot-1 stack.3 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
```

```
*00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9e:5c:76 3   3   Auto <none>             <none>          --EeMm-Nn --
 00:04:96:9c:53:b6 4   4   Auto <none>             <none>          --EeMm-Nn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured
```

The MAC Address column displays the factory MAC address for the node. The stack MAC address configuration information appears in the last three positions of the Flags column. As shown in the key at the bottom of the command display, the stack MAC configuration is displayed with the letters capital M, lower-case m, and lower-case i. If the flags read ---, the stack MAC address needs to be configured. If the flags read Mm-, the stack MAC address is already configured and in use.

2. To configure the stack to use the MAC address of the master node, follow these steps:
   a. Log in to the master node.
   b. Issue the command `configure stacking mac-address`.
   c. To verify that the command executed properly, enter the command `show stacking`.

      The Flags column should show `--i`, indicating that the stack MAC is configured but is not in use. After you restart the stack, the `i` will no longer appear in the Flags column.
   d. To verify that the stack MAC is configured consistently, enter the command `show stacking detail`.

      Verify that all of the stack MAC addresses are the same.
3. To configure the stack to use a MAC address from a non-master node, log in to the master console and enter the command `configure stacking {`**`node-address`** *`node-address`* | **`slot`** *`slot-number`*`}` **`mac-address`** .

   For example:

   ```
   Slot-1 stack.4 # configure stacking slot 2 mac-address
   This command will take effect at the next reboot of the specified node(s).
   ```

   The stack is configured to use the MAC address of the node in slot 2.
4. Reboot the stack.
5. Verify the new stack MAC address using the command `show stacking configuration`.

   For example:

   ```
   Slot-1 stack.5 # show stacking configuration
   Stack MAC in use: 02:04:96:9b:c1:34

   Node               Slot         Alternate          Alternate
   MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
   ------------------ --- --- ---- ------------------ --------------- --------- ---
   *00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
   ```

```
 00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9e:5c:76 3   3   Auto <none>             <none>          --EeMm-Nn --
 00:04:96:9c:53:b6 4   4   Auto <none>             <none>          --EeMm-Nn --
* - Indicates this node
```

Verifying the Configuration

To verify that your stack is configured as you intended, log into the master node and issue any or all of the following commands.

- show stacking on page 175
- show stacking configuration on page 176
- show slot on page 176
- show stacking detail on page 177
- show stacking stack-ports on page 178

These commands are also helpful when debugging problems with your stack.

Note
The examples in the following sections show a stack consisting of four switches.

show stacking

The `show stacking` command displays the stack topology along with each node's slot number, state, and role (master, backup, or standby).

```
Slot-1 Stack.1 # show stacking
Stack Topology is a Ring
Active Topology is a Ring
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:9c:e4:39  1     Active       Master   CA-
00:04:96:9b:c1:34  2     Active       Backup   CA-
00:04:96:9e:5c:76  3     Active       Standby  CA-
00:04:96:9c:53:b6  4     Active       Standby  CA-
* - Indicates this node
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology
```

In the command output, note that:

- The asterisk (*) that precedes the node MAC address indicates the node to which you are logged in. The node MAC address is the address that is factory assigned to the stackable switch.
- The slot number shown is the number currently in use by the related node. Because slot number configuration only takes effect during node initialization, a change in configured value alone does not cause a change to the slot number in use.
- If a node role has not yet been determined, the node role indicates <none>. In a ring topology, the node on which this command is executed is always the first node displayed. In a daisy chain, the ends of the daisy chain are the first and last nodes displayed.
- Even though the stack topology could be a ring, the active topology could be a daisy chain because it does not contain every node in the stack topology.

- If the node on which this command is being executed is not active, the stacking topology is replaced with a line similar to this one:
  ```
  This node is not in an Active Topology.
  ```
- It is possible for a node to be in stabilizing or waiting state and still be in the active topology.

show stacking configuration

The `show stacking configuration` command displays a summary of the stacking configuration for all nodes in the stack.

```
Slot-1 Stack.3 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
00:04:96:9e:5c:76 3   3   Auto <none>             <none>          --EeMm-Nn --
00:04:96:9c:53:b6 4   4   Auto <none>             <none>          --EeMm-Nn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured
```

In the command output, note especially the values in the **Flags** column:

- All nodes should have the `Ee`, `Mm`, and `Nn` flags active.
- Additionally, the master and backup nodes should have the `Cc` flags active.

(The meanings of the flags are displayed at the bottom of the table in the command output.)

show slot

The `show slot` command shows the states of the nodes as they move from the empty to operational state.

```
Slot-1 Stack.2 # show slot
Slots    Type                 Configured          State       Ports
-------------------------------------------------------------------
Slot-1   5520-12MW-36W                            Operational   48
Slot-2   5520-12MW-36W                            Operational   48
Slot-3   5520-12MW-36W                            Operational   48
Slot-4   5520-12MW-36W                            Operational   48
Slot-5                                            Empty          0
Slot-6                                            Empty          0
Slot-7                                            Empty          0
Slot-8                                            Empty          0
```

show stacking detail

The `show stacking {`**`node-address`** *`node_address`* | **`slot`** *`slot_number`*`}` **`detail`** command displays a full report about a particular node.

```
Slot-1 Stack.33 # show stacking slot 1 detail
Stacking Node 00:04:96:26:6b:ec information:
Current:
Stacking : Enabled
Role : Master
Priority : Automatic
Slot number : 1
Stack state : Active
Master capable? : Yes
Stacking protocol : Enhanced
License level restriction : <none>
In active topology? : Yes
Factory MAC address : 00:04:96:26:6b:ec
Stack MAC address : 02:04:96:26:6b:ec
Alternate IP address : <none>
Alternate gateway : <none>
Stack Port 1:
State : Operational
Blocked? : No
Control path active? : Yes
Selection : Alternate (23)
Stack Port 2:
State : Operational
Blocked? : Yes
Control path active? : Yes
Selection : Native
Configured:
Stacking : Enabled
Master capable? : Yes
Slot number : 1
Stack MAC address : 02:04:96:26:6b:ec
Stacking protocol : Enhanced
License level restriction : <none>
Stack Port 1:
Selection : Alternate (23)
Stack Port 2:
Selection : Native
```

In the command output, note that:

- If you issue `show stacking detail` without a *`node-address`* or *`slot`* parameter, the output is generated for all nodes in the stack topology.
- The *`slot`* parameter is available only in stacking mode. The *`node-address`* parameter is always available.
- Current information represents stacking states and configured values that are currently in effect. Configured information is that which takes effect at node reboot only.
- The node's role can be Master, Backup, Standby, or none.
- License level restrictions are Edge, Advanced Edge/Base, or Core/Premier.

show stacking stack-ports

The `show stacking stack-ports` command displays the states of each stacking port in the stack. This information can help you verify that all ports are properly connected and operational.

```
Slot-1 Stack.5 # show stacking stack-ports
Stack Topology is a Ring
Slot Port Select Node MAC Address  Port State  Flags Speed
---- ---- ------ ----------------- ----------- ----- -----
*1   1    27     00:04:96:9c:e4:39 Operational C-      10G
*1   2    28     00:04:96:9c:e4:39 Operational CB      10G
2   1    27     00:04:96:9b:c1:34 Operational CB      10G
2   2    28     00:04:96:9b:c1:34 Operational C-      10G
3   1    15     00:04:96:9e:5c:76 Operational C-      10G
3   2    16     00:04:96:9e:5c:76 Operational C-      10G
4   1    15     00:04:96:9c:53:b6 Operational C-      10G
4   2    16     00:04:96:9c:53:b6 Operational C-      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is Blocked
```

In the command output, a state other than `Operational` indicates a potential problem. Check the physical connections and the software configuration for the ports in question. For more information, see Troubleshooting a Stack on page 211.

Using the Mode Button to Automatically Stack

The **mode** button is a front-panel button used to select alternate front-panel LED display modes that convey additional port and stacking information. On all ExtremeSwitching Universal platforms, this button can be used to initiate automatic stacking.

Automatic stacking using the **mode** button feature is aimed at zero-touch customers who want to deploy stacking in remote sites. You can ship switches to remote sites and have a local (non-technical) installer mount the switches in the rack, attach the power cables, and connect the stacking cables. The installer can then press the **Mode** button to initiate the automatic stacking sequence. After stacking is complete, the switches can be provisioned and managed remotely using a cloud connection.

You can stack 1 to 8 nodes. If you have only a single node in the stack, stacking occurs in a daisy-chain topology. If there is more than a single node in the stack topology, the stack is connected as a ring. If a ring is not detected, the automatic stacking sequence does not run when you presses the **Mode** button.

You can build a stack a node at a time by initiating automatic stacking on one switch that, after rebooting, becomes the primary node. Attaching nodes subsequently causes those nodes to join the stack.

Important

To use automatic stacking, each switch must have the factory default configuration. If any of the switches do not have the factory configuration (directly out of the shipping box), remove any configuration and restore the factory defaults by running the command `unconfigure switch {all | erase [all | nvram]}` with the all option.

License level and ExtremeXOS version mismatches are automatically reconciled. However, license mismatches due to installing feature licenses cannot be resolved automatically. The applicable node is left in the failed state.

To use automatic stacking, all switches must:

- Have the same license level.
- Be running ExtremeXOS 30.3 or later.

  If the primary node is running a newer version of ExtremeXOS than the nodes, the nodes are upgraded to the newer ExtremeXOS version.
- Not have an Extended Edge Switching topology.

Note

A mixed stack of ExtremeSwitching 5720 and 5520 switches cannot use automatic stacking.

To begin automatic stacking:

1. Remove the switches from their boxes and mount the desired number of switches in the desired locations. For information about physically setting up switches, see *ExtremeSwitching Quick Reference Guide* for the switch series.
2. Attach the stacking cables to form a ring between the switches (see Ring Topology: Recommended for Stacking on page 149).
3. Attach the power cables to each switch, and then plug them into a power source.
4. Wait for all switches to boot (approximately 2 minutes).
5. Verify that there is **LINK UP** on all stack ports by checking the stack port LEDs.

   Note

   A single node stack (no cabling) will not have a lit LED.

6. On the switch that is going to be the stack primary, push the **Mode** button twice and verify that the **STK** LED is solid green.
7. Press and hold the **Mode** button for at least 5 seconds.

AFter releasing the **Mode** button (following the final step), the front panel port LEDs (on the primary node) flash in an alternating pattern followed by a delayed reboot of all of the switches. The LEDs flash for approximately 15 seconds prior to reboot. After the reboot completes, the switches are stacked.

If the front panel port LEDs on the primary node do *not* flash (flashing LEDs indicate that automatic stacking has been initiated), try the following:

- Check the cabling for correct ring topology, and fix as needed (see Ring Topology: Recommended for Stacking on page 149).
- Check that the stack port **LINK UP** LEDs are lit (see step 5 on page 179).

Note
A single node stack (no cabling) will not have a lit LED.

If you can verify that both the cabling and the **LINK UP** LEDs are lit, then further troubleshooting requires accessing the switch with a console or by cloud access to check the switch's error log.

Using a USB Flash Drive to Automatically Stack

A USB flash drive can be used for auto-stacking on switches with or without a **mode** button.

You can stack 1 to 8 nodes. If you have only a single node in the stack, stacking occurs in a daisy-chain topology. If there is more than a single node in the stack topology, the stack is connected as a ring. If a ring is not detected, the automatic stacking sequence does not run when you insert the USB flash drive.

You can build a stack one node at a time by initiating automatic stacking on one switch that, after rebooting, becomes the primary node. Attaching nodes subsequently causes those nodes to join the stack.

Important
To use automatic stacking, each switch must have the factory default configuration. If any of the switches do not have the factory configuration (directly out of the shipping box), remove any configuration and restore the factory defaults by running the command `unconfigure switch {all | erase [all | nvram]}` with the all option.

License level and ExtremeXOS version mismatches are automatically reconciled. However, license mismatches due to installing feature licenses cannot be resolved automatically. The applicable node is left in the failed state.

To use automatic stacking, all switches must:

- Have the same license level.
- Be running ExtremeXOS 31.4 or later.

  If the primary node is running a newer version of ExtremeXOS than the nodes, the nodes are upgraded to the newer ExtremeXOS version.
- Not have an Extended Edge Switching topology.

To begin automatic stacking:

1. Remove the switches from their boxes and mount the desired number of switches in the desired locations. For information about physically setting up switches, see *ExtremeSwitching Quick Reference Guide* for the switch series.
2. Attach the stacking cables to form a ring between the switches (see Introduction to Stacking on page 146).
3. Attach the power cables to each switch, and then plug them into a power source.
4. Wait for all switches to boot (approximately 2 minutes).
5. Verify that there is **LINK UP** on all stack ports by checking the stack port LEDs.

   Note
   A single node stack (no cabling) will not have a lit LED.

6. On the switch that is going to be the stack primary, insert the USB flash drive.

The USB flash drive must contain a file named “autoStackingMasterNode,” which invokes the auto-stacking logic. The front panel port LEDs (on the primary node) flash in an alternating pattern followed by a delayed reboot of all of the switches. The LEDs flash for approximately 15 seconds prior to reboot. After the reboot completes, the switches are stacked.

Note
The delay for auto-stacking can be up to 5 minutes.

If the front panel port LEDs on the primary node do *not* flash within the 5 minute timeout period (flashing LEDs indicate that automatic stacking has been initiated), try the following:

- Check the cabling for correct ring topology, and fix as needed (see Introduction to Stacking on page 146).
- Check that the stack port **LINK UP** LEDs are lit.

  Note
  A single node stack (no cabling) will not have a lit LED.

If you can verify that both the cabling and the **LINK UP** LEDs are lit, then further troubleshooting requires accessing the switch with a console or by cloud access to check the switch's error log.

Auto-Discovery for Universal Hardware

Auto-Discovery automatically detects cable types inserted into the U1 and U2 stacking port, and adjusts the configured speed of the port. Detection of stacking cables is performed during startup of the switch. Auto-Discovery is supported on ExtremeSwitching 5420 and 5520 series switches beginning in ExtremeXOS 31.5.

Auto-discovery operates on stack ports U1 and U2 on both 5420 and 5520 series switches. On 5420 series switches, the stack ports support either SFP+ or SFP-DD type cables. On 5520 series switches, the stack ports only support QSFP+ type cables.

For example, if a 5420 series switch is configured for Native V40 stacking and Auto-Discovery detects an SFP-DD cable present in either the U1 or U2 stack ports during start up, the stack port speed is automatically reconfigured for Native V80 (to match the speed of the SFP-DD cable).

Important
Changing the stacking cables on an operational stack to achieve a different stack speed requires a reboot.

Supported Platforms

Auto-Discovery for Universal Hardware is supported on ExtremeSwitching 5420 and 5520 Series switches.

CLI Command

Auto-Discovery is enabled by default on the ExtremeSwitching 5420 and 5520 Series switches. Use the `configure` **`stacking-support auto-discovery`** `[`**`disable`**`]` command to disable Auto-Discovery for Universal Hardware.

Note
Auto-Discovery is automatically re-enabled whenever the Universal Hardware is restored back to the factory configuration when the **`unconfigure switch all`** command is used.

Limitations

The following limitations apply to Auto-Discovery for Universal Hardware:

- Stacking-support must be enabled in order for the feature to execute.
- The feature will terminate if you have configured Alternate stacking on the ExtremeSwitching 5520.
- The feature can be disabled by entering **`configure stacking-support auto-discovery disable`**

Managing an Operational Stack

The following topics describe common tasks for logging into an operational stack and managing it.

Logging into a Stack

You can log into any node in a stack, but you can control more stack features when you log into the master. The following guidelines describe the options available to you when you log into different nodes:

- On master nodes, all features supported by the switch license operate correctly.
- On backup nodes, most show commands show correct data for the active stack. For example, `show vlan {virtual-router vr-name}` shows all configured VLANs.
- On all non-master nodes, most of the configuration commands are rejected. However, the failsafe account, enable license, and stacking configuration commands work on any node.
- On standby nodes, most show commands do not show correct data for the current stack operation. However, the `show switch {detail}`, `show licenses`, and all `show stacking` commands show correct data.
- If a node is connected to the stack and stacking is not enabled, you can still configure stacking features on that node.

The login security that is configured on the master node applies when logging into any node in the active topology. This includes any active node that is present in a slot. A node that is disabled for stacking is its own master, and uses its own security configuration.

You can log in to a SummitStack node using the following methods:

- Console connection to any node
- Management connection to the master
- Management connection to a standby node
- Telnet session over the stack from any active node to any other node in the same active topology

Logging in Through the Console Port

You can use the console port on any switch to manage the stack.

If you connect to the master node, you can configure and manage the stack. If you connect to a non-master node, you can view node status and configure only a few options from the node to which you are connected. However, you can use the Telnet feature to connect to another node and manage that node as if you were connected to it (see Logging Into a Node From Another Node on page 184).

Logging in from the Management Network

The management network is an Ethernet network to which the management port of each switch connects.

The primary management IP address is assigned to the master node. You can use a terminal emulation program and this IP address to connect to the master for configuration and management.

The alternate management IP addresses allow you to connect to individual nodes from your management network. During normal operation, you connect to the stack using

the primary management IP address. However, if the stack is split, you can use the alternate management IP address to connect to the other half of the stack. For more information, see Configuring an Alternate IP Address and Gateway on page 192.

After you log in to a master or standby node through the management network, you can Telnet to any other node and control that node as if you were directly connected to it. For more information, see Logging Into a Node From Another Node on page 184.

Logging Into a Node From Another Node

You may log into any node in the active topology from any other node in the same active topology. If you do not know the slot number of the node to which you want to connect, enter the `show slot` command. You can Telnet to any switch that appears in the `show slot` command display.

Note
If the node to which you want to connect does not appear in the `show slot {slot {detail} | detail }` command display, you can connect to the node through the its console port or management port.

You have the most control over the stack when you log in to the master.

1. Determine which node is the master using the command `show stacking`.
2. Telnet to another node using the command `telnet slot slot-number` .

   The switches must be active in the stack for this command to function.

You are not prompted for your username or password. You are logged in to the same account (with corresponding rights) with which you accessed the originating slot.

When the Telnet program accepts a connection from another node in the stack, it performs security validation. The master node validates all login security information (except for the failsafe account), regardless of the node into which you are attempting to establish a login. If you are not able to log in using your user credentials, use the failsafe account to log in.

Managing Licenses on a Stack

The SummitStack feature is not licensed separately. You can use the SummitStack feature any license.

The rules for licensing are as follows:

- The effective license level of all master-capable nodes must be the same.
- If you set a license level for the stack, then each node must be at that license level or higher.

- If you do not set a license level for the stack, the license level the stack uses is the effective license level of the node that is elected master at startup.

  Note
  If the stack is using a certain license level, and you attempt to add a master-capable node that is using a lower level license, the node does not become operational. In response to the `show slot {slot {detail} | detail }` command, the node displays as Failed with a License Mismatch reason.

- Master-capable nodes continuously monitor the effective license level of the master node.
- Nodes with higher license levels than other nodes can be restricted to operate at a lower or effective license level.
- With Extreme Platform ONE, a single license is applied to all nodes in the stack, and individual license counts are managed by the cloud.

Viewing Switch Licenses and License Restrictions

1. To view the current license information for the stack, log into that master node and enter the `show licenses {[slot slot |all]} {detail}` command with the all option.
   The command displays information similar to the following:

```
# show licenses slot all
 Slot Enabled License          Effective License        Feature Packs
 ---- ------------------------ ------------------------ -------------------------
 1    Base                     Base                     None
 2    Premier                  Base                     MACsec
```

   The Enabled License Level is the purchased license level. This is the maximum level at which this node can operate without purchasing a license level upgrade.

   The Effective License Level is the operating license level. If a license level restriction is configured for this node, the effective license level may be lower than the enabled license level. All master-capable switches must be operated at the same effective license level.

   Feature Packs shows which feature packs are installed on that node.
2. On the master node, run `show stacking configuration` to view the license level restrictions configured for all nodes in a stack.

```
Slot-1 stack.2 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9e:5c:76 3   3   Auto <none>             <none>          --EeMm-Nn --
 00:04:96:9c:53:b6 4   4   Auto <none>             <none>          --EeMm-Nn -e
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
```

```
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (B) Base, or (P) Premier in use,
        (b) Base, or (p) Premier configured,
        (-) Not in use or not configured
```

License level restrictions appear in the **Lic** column. The license level restriction in use appears first, represented by a capital letter as shown in the display legend. The configured license level restriction appears second, represented by a lower-case letter. When the letters in the Lic column are different, for example `-e` or `Ae`, the node is configured with a different license level restriction than the one that is currently in use.

To put the configured license level restriction into effect, you must reboot the node.

Enabling a Switch License

All nodes must have a purchased license level at least equal to the license level of the master node in order to become operational in the stack. The purchased license level of a node can be enabled only after you log in to that node.

For instructions on enabling a license on a node, see Software Upgrade and Boot Options on page 1697. With Extreme Platform ONE, a single license is applied to all nodes in the stack, and individual license counts are managed by the cloud.

Restricting a Switch License Level

If the master-capable nodes in a stack have different license levels and you want to operate a stack at the minimum license level, you can apply a license level restriction. The restriction is stored in the NVRAM of each master-capable node. It forces the node to reduce its license level below its purchased level at node restart time for the life of the restart. This reduced license level is called the effective license level and can be displayed by entering the `show licenses` command on the node you want to evaluate.

To restrict a master-capable node to operate at a license level that is lower than the one purchased for the node, use the command:

`configure stacking {`**`node-address`** *`node-address`* | **`slot`** *`slot-number`*`}` **`license-level`** *`license_restriction`*.

Note
If using Extreme Platform ONE, restricting license levels in or for a stack is not necessary.

In the following example, node 7 is restricted to operate at the Edge license level:

```
Slot-1 Stack.3 # configure stacking slot 7 license-level edge
This command will take effect at the next reboot of the specified node(s).
```

You must reboot the master-capable node for the command to take effect.

The command restricts the specified node to operate at the specified license level. The specified license level must match the effective license level of all master-capable nodes in the stack. To avoid stack reboots when future license level upgrades are purchased, during initial deployment you should purchase the same license level for

every master-capable node in the stack, and the license level restriction should not be configured.

Upgrading Stack Licenses

You can purchase license level upgrades for ExtremeSwitching switches. All master-capable switches in a stack must run the same license level. If the license you want to run is not available for a specific switch, you cannot use that switch and that license level as a master-capable switch. For example, if you want to upgrade to the core license, your master-capable nodes must be switches that support the core license.

Note
For information about which switches support which licenses, see the Switch Engine v33.6.1 Licensing Guide document. That document also lists which switches support the SummitStack feature.

To upgrade the licenses for the switches in a stack, follow these steps:

1. Log in to the master node.
2. Enter the `show stacking` command and note the role (master, backup, or standby) of each node in the stack.
3. Enter the `show stacking configuration` command and note any nodes that are configured with a license level restriction.

   See Viewing Switch Licenses and License Restrictions on page 185 for more information.
4. Install the required license level in each master-capable node (backup and standby nodes) by logging into each node (`telnet slot` *`slot-number`*) and entering the command:

   `enable license {`**`software`**`}` *`key`*
5. For universal stacked switches, you do not need to log into each node for license installation. You can install licenses on each slot from the master slot by entering the command:

   `install license` **`file`** *`filename`* **`slot`** *`slot`*
6. Enter the license key given to you by Extreme Networks when you purchased the upgrade.
7. Use the commands in Step 4 to install the required license level on the master node.
8. If any nodes are configured with a license level restriction that is lower than the intended operating license level of the stack, log into the master node and remove the stack license level restriction using the command:

   `unconfigure stacking license-level`

   This command removes the restriction on all nodes.
9. If you removed a license level restriction, reboot the stack to put the license level restriction removal into effect using the command:

   `reboot {[`**`time`** *`mon day year hour min sec`*`] |` **`cancel`**`} {`**`slot`** *`slot-number`* `|` **`node-address`** *`node-address`* `|` **`stack-topology`** `{`**`as-standby`**`}}`

10. Verify that all master-capable nodes are operating at the intended license level.

    On the master node, run `show licenses` and `show slot {`*`slot`* `{`**`detail`**`} | `**`detail`** `}`.

    If no slot shows as Failed, then all master-capable nodes are operating at the effective license level shown for the master node.

Upgrading ExtremeXOS on a Stack

The following topics describe how to upgrade the ExtremeXOS software and the bootrom on the switches in a stack.

Stack Rolling Software Upgrade

A Stack Rolling Software Upgrade (Rolling Upgrade) enables a stack to continue functioning during a software upgrade. The benefit of a Rolling Upgrade is minimal impact to the devices connected via a LAG or other multi-homing technologies that allow a device to be simultaneously connected to different stack nodes. Performing a Rolling Upgrade does not change the current method of installing software, only how the stack is rebooted.

The Rolling Upgrade process reboots all the Standby nodes, reboots the Backup node, and then initiates a failover to make the old Backup node the new Primary. The failover results in the old Primary node rebooting. Since the selected software version on each node is the newer, upgraded software version, after each node reboots, they will run the new software version.

Note
The rolling reboot upgrade will work only when performed across patch versions of a software release. It might not work as expected when used to upgrade the stack across software versions.

Performing a Rolling Upgrade on page 188 contains steps on initiating a Rolling Upgrade.

Performing a Rolling Upgrade

To begin a Rolling Upgrade, all nodes in the stack must have the same selected software version.

Complete these steps to perform a Rolling Upgrade.

Note
Performing a Rolling Upgrade does not change the current method of installing software, only how the stack is rebooted.

1. Download the software image using the **`download`** command.
2. Install the image (if not already completed in the previous step) using the **`install`** command.
3. Reboot the stack using the **`reboot`** command, specifying the `reboot rolling` option.

A failover is performed to complete the upgrade.

If a Rolling Upgrade fails, the nodes in the stack will likely be in different stages of the rolling upgrade process. Any nodes that have started the reboot process will be running the newer software version. Nodes that did not begin rebooting will still be running the older software. The stack should not be left in this condition for an extended period of time.

Possible remedies include:

- Reverting a Rolling Upgrade on page 189
- Finishing a Rolling Upgrade on page 190

Reverting a Rolling Upgrade

To revert an upgrade, the Rolling Upgrade process is performed in reverse order by completing the following steps:

1. Determine which partition is currently selected on each node by entering the **show slot detail** command.

   The include keyword can be used to simplify the output. In the following example, the Backup and both Standy nodes have been rebooted and are running the newer software version:

```
# show slot detail | include "information|Image|State|ver:"
Slot-1 information:
     State:               Operational
     Current State:       MASTER
     Image Selected:      primary
     Image Booted:        secondary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
Slot-2 information:
     State:               Operational
     Current State:       BACKUP (In Sync)
     Image Selected:      primary
     Image Booted:        primary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
Slot-3 information:
     State:               Operational
     Current State:       STANDBY
     Image Selected:      primary
     Image Booted:        primary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
Slot-4 information:
     State:               Operational
     Current State:       STANDBY
     Image Selected:      primary
     Image Booted:        primary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
```

2. Change the selected software partition to the old version on all nodes by entering the **use image secondary** command.

3. When all nodes have the old software version selected, reboot each node running the new software version, starting with the Backup node first.

   When the Backup node reboots, it should go back to “In-Sync” with the Primary node.

   Note
   After each reboot, allow a few minutes for the network to converge before proceeding.

4. Reboot any Standby nodes individually, ensuring that the rebooted node is operational before moving on to the next node.

When all nodes are running the old software version, the upgrade is reverted. At this point, all nodes should have both their selected and booted partitions set to the partition that contains the old software version.

Finishing a Rolling Upgrade

To manually finish an upgrade and continue rebooting nodes that were not rebooted, complete the following steps:

1. Determine which partition was booted for each node by entering the **show slot detail** command.

   The include keyword can be used to simplify the output. In the following example, only one of the Stanby nodes (Slot-3) was rebooted and is running the newer software version:

```
# show slot detail | include "information|Image|State|ver:"
Slot-1 information:
     State:               Operational
     Current State:       MASTER
     Image Selected:      primary
     Image Booted:        secondary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
Slot-2 information:
     State:               Operational
     Current State:       BACKUP (In Sync)
     Image Selected:      primary
     Image Booted:        secondary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
Slot-3 information:
     State:               Operational
     Current State:       STANDBY
     Image Selected:      primary
     Image Booted:        primary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
Slot-4 information:
     State:               Operational
     Current State:       STANDBY
     Image Selected:      primary
     Image Booted:        secondary
     Primary ver:         31.3.1.22
     Secondary ver:       31.3.0.106
```

2. Reboot any Standby nodes running the old software version individually, ensuring that the rebooted node is operational before moving on to the next node.

3. When all the Standby nodes are running the new software version and the network has converged, reboot the Backup node and allow it to go back “In-Sync” with the Primary node.

   Note
   After each reboot, allow a few minutes for the network to converge before proceeding.

4. Failover control of the stack from the Primary node to the Backup node by entering the **run failover** command.

When all nodes are running the newer software version, the upgrade is finished. At this point, all nodes should have both their selected and booted partitions set to the partition that contains the newer software version.

Upgrading the Software on all Active Nodes

You can centrally upgrade the software on all active nodes in a stack. To upgrade all nodes in the stack, all nodes must be running an ExtremeXOS release that supports stacking.

Note
You cannot automatically update a switch running ExtremeXOS 30.2 or earlier to ExtremeXOS 30.3 or later due to a file system compatibility issue. If you are inserting a switch that has ExtremeXOS 30.2 or earlier, prior to inserting the switch into the stack topology, you need to manually upgrade the switch with the following procedure.

Hitless upgrade is not supported in a stack, except during a Stack Rolling Software Upgrade.

1. To download and install a new image, the active partitions (primary or secondary) of all non-master nodes must match the active partition of the master node.
   a. To determine the active partition selected on all nodes and the ExtremeXOS versions installed in each partition, use the `show slot {slot {detail} | detail }` command with the detail option. If the node being upgraded is running on the primary partition, then the new image is downloaded and installed on the secondary partition.
   b. If the active partition is different on some nodes, the action you take depends on what is stored in both partitions:

      If both primary and secondary partitions have the same ExtremeXOS release, you may use the following commands to cause a node to use the same active image as the rest of the stack:

      ```
      use image {primary | secondary} slot slot-number
      reboot slot slot-number
      ```

2. Download a new ExtremeXOS software release and install it on all nodes on the active topology using the command: `download [url url {vr vrname} | image [active | inactive] [[hostname | ipaddress] filename {{vr} vrname} {block-size block_size}] {partition} {install {reboot}}`

3. Restart all nodes in the new release using `reboot {[time mon day year hour min sec] | cancel} {slot slot-number} | node-address node-address | stack-topology {as-standby} | all} | rolling}`

Upgrading the Software on a Single Node

To upgrade the software on a single active node:

1. Enter the following commands to download an image to a node:

   ```
   download image [[hostname | ipaddress] filename {{vr} vrname} {block-size block_size}] {partition} {install {reboot}}

   use image {partition} [primary | secondary] slot slotid

   reboot slot slot number
   ```

   The slot number is the one in use by the active node that is to be upgraded. Be sure that you keep the same image versions on all the other nodes as you have on the master node.

   Alternatively, if your master node has the same image versions in its partitions that you want installed in the node to be upgraded, you can use the command `synchronize {slot slotid}` to upgrade both images and select the desired image.

2. You can upgrade the image on an active node even if the node shows as Failed when using the `show slot` command.

Upgrading the Bootrom

The SummitStack feature does not require a bootrom upgrade. You should not upgrade the bootrom of any node unless there are specific reasons to do so. However, the SummitStack feature does allow centralized bootrom upgrade.

You can download and install the bootrom to a specific slot using the slot parameter. The slot parameter is available only on stackable switches in the active stack topology. For information on upgrading the bootrom, see Software Upgrade and Boot Options on page 1697.

If you do not provide a slot number, the stack attempts to download the bootrom image and install it on all stackable switches in the active topology.

Configuring an Alternate IP Address and Gateway

The stack has a primary IP address and subnetwork. These values are assigned using the `configure vlan mgmt ipaddress` command. There can also be static or default routes associated with the stack.

For each node in the stack, you can configure an alternate management IP address, subnetwork mask, and default gateway. The alternate IP address is restricted to being a member of the primary IP subnetwork that is configured on the management *VLAN*, and thus the alternate IP subnetwork must exactly match the primary IP management subnetwork. A subnetwork match is exact if the subnetwork portion of the IP addresses

match exactly. For example, 10.11.12.1/24 and 10.11.12.2/24 are an exact subnetwork match (because both represent the subnet 10.11.12.0/24).

Standby nodes always install their configured alternate management IP address and gateway on the management interface. A standby node does not have the ability to verify whether the configured alternate IP address matches the primary management IP subnetwork of the stack.

The backup and master nodes have the ability to verify the configured alternate IP address. The master and backup nodes compare the primary IP subnetwork information to the alternate IP subnetwork. If there is a match, the backup node installs the primary IP management subnetwork's default routes and installs only the alternate management IP address (not the primary IP address). The master node installs both the configured management subnetwork with specific IP address and the alternate IP address. In this case, the alternate gateway is not used, expecting that primary management routes are configured or will be configured. In either case, if the alternate IP subnetwork does not match the configured management subnetwork, the alternate IP address is not installed on the management interface.

Each node in the stack normally installs its alternate IP address on the management subnetwork. When an ARP request for the alternate IP address is satisfied, the stackable switch supplies its factory-assigned MAC address and not the stack MAC address. Only the master node installs the primary IP address. An ARP request for the configured management IP address returns the configured stacking MAC address. Because of the above behavior, all nodes are reachable over their management ports even during a dual master condition. The VLAN used is the management VLAN (VID 4095) and is untagged.

The alternate gateway is only installed on a master or backup node when the primary management IP subnetwork is not configured. Once the primary IP subnetwork is installed, the alternate gateway is removed. The alternate gateway is always installed on a standby node.

If a dual master condition occurs because a stack has been severed, the alternate IP addresses and associated MAC addresses are unique, and it is possible to use telnet or ssh to reach any node. Any node on the segment with the incorrect master can then be used to reboot the entire stack segment into standby mode if you want to rejoin the stack segments later.

If a node is operating in stacking mode, the alternate management IP address configuration takes effect immediately.

Note
Only IPv4 alternate management IP addresses are supported in this release.

To configure an alternate IP address and gateway, follow these steps.

1. View the alternate IP address configuration using the `show stacking configuration` command:

```
Slot-1 stack.3 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
```

```
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9e:5c:76 3   3   Auto <none>             <none>          --EeMm-Nn --
 00:04:96:9c:53:b6 4   4   Auto <none>             <none>          --EeMm-Nn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured
```

In the preceding example, no alternate IP address or alternate gateway is configured.

2. If you have a continuous block of IP addresses to assign to the stack, enter the `configure stacking alternate-ip-address [`*ipaddress netmask* | *ipNetmask*`] gateway` **automatic** command.

   For example:

```
Slot-1 Stack.4 # configure stacking alternate-ip-address 10.127.4.131/24 10.127.4.254
automatic
Slot-1 Stack.5 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto 10.127.4.131/24    10.127.4.254    CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto 10.127.4.132/24    10.127.4.254    CcEeMm-Nn --
 00:04:96:9e:5c:76 3   3   Auto 10.127.4.133/24    10.127.4.254    --EeMm-Nn --
 00:04:96:9c:53:b6 4   4   Auto 10.127.4.134/24    10.127.4.254    --EeMm-Nn --
* - Indicates this node
```

3. If you do not have a continuous block of IP addresses for the stack, assign an alternate IP address and gateway to each node using the `configure stacking [`**node-address** *node-address* | **slot** *slot_number*`] alternate-ip-address [`*ipaddress netmask* | *ipNetmask*`] gateway` command.

   For example:

```
Slot-1 Stack.6 # configure stacking slot 4 alternate-ip-address 10.127.4.139/24
10.127.4.254
```

Note

If you try to assign an alternate IP address and gateway to a node that is already configured with these parameters, an error message appears. To remove an existing configuration so you can change the alternate IP address and gateway, enter the `unconfigure stacking {`**node-address** *node_address* | **slot** *slot_number*`}` **alternate-ip-address** command.

4. Enter the `show stacking configuration` command to verify that the alternate IP address and gateway is configured as intended for each node.

```
Slot-1 Stack.7 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
```

```
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway          Flags     Lic
------------------ --- --- ---- ------------------ ---------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto 10.127.4.131/24    10.127.4.254     CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto 10.127.4.132/24    10.127.4.254     CcEeMm-Nn --
 00:04:96:9e:5c:76 3   3   Auto 10.127.4.133/24    10.127.4.254     --EeMm-Nn --
 00:04:96:9c:53:b6 4   4   Auto 10.127.4.139/24    10.127.4.254     --EeMm-Nn --
* - Indicates this node
```

Viewing the Alternate IP Address

To view the alternate IP address for a node, run `show vlan mgmt` or `show ipconfig mgmt`.

show vlan mgmt Command

The `show vlan mgmt` command shows the alternate management IP address as applied to the management *VLAN* on the local unit. This allows you to see how the configured alternate management IP address has been applied.

The `show vlan mgmt` command displays the following information:

```
Slot-1 Stack.1 # show vlan "Mgmt"
VLAN Interface with name Mgmt created by user
    Admin State:         Enabled     Tagging:   802.1Q Tag 4095
    Description:         Management VLAN
    Virtual router:      VR-Mgmt
    IPv4 Forwarding:     Disabled
    IPv4 MC Forwarding:  Disabled
    IPv6 Forwarding:     Disabled
    IPv6 MC Forwarding:  Disabled
    IPv6:                None
    STPD:                None
    Protocol:            Match all unfiltered protocols
    Loopback:            Disabled
    NetLogin:            Disabled
    OpenFlow:            Disabled
    QosProfile:          None configured
    Flood Rate Limit QosProfile:       None configured
    Ports:   1.           (Number of active ports=1)
     Untag: Mgmt-port on Mgmt-1 is active
```

For the management VLAN, a secondary address cannot be configured and so the Secondary IP line does not appear.

The Alternate IP line shows one of the following:

- The configured alternate management IP address if it has been activated.
- <none> if it has not been configured.
- Mismatch if it has been configured but does not exactly match the Primary IP subnet.

show stacking configuration Command

The `show stacking configuration` command shows how the nodes are configured in a stack topology. The configured values shown are the ones actually stored in the remote nodes at the time you issue this command.

Some stacking parameters do not take effect until the next restart, the configured values and the values currently being used are both shown. Specifically, this applies to the slot number, whether or not stacking is enabled, the master-capable configuration, the license level restriction, and the stack MAC configuration.

The only parameters that take effect without a reboot are the node priority and the alternate management IP subnetwork and gateway.

The following example shows the stacking configuration for a switch:

```
Slot-1 Stack.2 # show stacking configuration
Stack MAC in use: 02:04:96:26:6b:ed
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:26:6b:ed 1   1   Auto <none>             <none>          CcEeMm--- --
00:04:96:34:d0:b8 2   2   Auto <none>            <none>          CcEeMm--- --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
(E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
(M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
(N)  Stack link protocol Enhanced in use, (n) Stack link protocol Enhanced configured,
(i) Stack MACs configured and in use are not the same or unknown,
(-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
(c) Core, (a) Advanced edge, or (e) Edge configured,
(-) Not in use or not configured
```

For the management VLAN, a secondary address cannot be configured and so the Secondary IP line does not appear.

The Alternate IP Address line shows one of the following:

- The configured alternate management IP address if it has been activated.
- <none> if it has not been configured.
- Mismatch if it has been configured but does not exactly match the Primary IP subnet.

Viewing Stacking Port Statistics

To view the status of any stacking port, run any of the following variations:

`show ports stack-ports` *`stacking-port-list`* `utilization {`**`bandwidth`** `|` **`bytes`** `|` **`packets`**`}`

`show ports {`**`stack-ports`** *`stacking-port-list`* `|` *`port_list`*`} statistics {`**`norefresh`**`}`

`show ports {`*`port_list`* `|` **`stack-ports`** *`stacking-port-list`*`} rxerrors {`**`norefresh`**`}`

`show ports {`**`stack-ports`** *`stacking-port-list`* `|` *`port_list`*`} txerrors {`**`norefresh`**`}`

The commands accept stacking port ranges that span multiple nodes. For example, both *`port-list`* and *`stacking-port-list`* might be expressed as `1:1-3:2`.

There are no stacking port configuration options.

See Switch Engine v33.6.1 Command References for details about these commands.

Note
You can not disable a port that is configured to stack.

Configuring Master-Capability

Each node is configurable to be master-capable or not. This means that a node can either be allowed to take on any node role, or be restricted to executing the standby node role only. The default is that a node can take on any role. The restriction is used to avoid the dual master condition. A master-capability configuration change takes effect at the next restart.

You can configure one or more nodes to be allowed to operate either as a master or a backup.

The `configure stacking master-capability` command allows you to set the master-capability of specific nodes, while configure stacking redundancy allows you to set the master-capability on all nodes in the stack.

The commands do not allow you to disable master-capability on all nodes in a stack topology.

Note
If the entire stack is restarted in stacking mode without any node having master capability, you need to know the failsafe account and password to log into any node in the stack. If you do not know the failsafe account information, you might need to rescue the stack. See Rescuing a Stack that has No Primary-Capable Node on page 216.

You can use any of the following commands to configure the master-capability:

- `configure stacking [`**`node-address`** *`node_address`* `|` **`slot`** *`slot_number`*`] `**`master-capability`** `[`**`on`** `|` **`off`**`]`
- `configure stacking redundancy [`**`none`** `|` **`minimal`** `|` **`maximal`**`]`

Rebooting a Stack

You can reboot a stack by entering the command `reboot` from the master. You can:

- Reboot all the nodes in the stack topology by entering the `reboot` command from a master node login session.
- Reboot a specific node by entering the `reboot node-address` *`node-address`* command.
- Reboot all nodes in the active topology by entering the `reboot stack-topology` command.
- Move a node to a standby node.

- Reboot the stack topology so that every node comes up in standby role.
- Reboot an active node from another active node by entering the `reboot slot slot-number` command.
- Reboot the stack by rolling through each node individually, allowing multi-homed devices to remain connected through the reboot by entering the `reboot` **`rolling`** command.

Changing the Stack Configuration

The following topics describe common tasks for changing the configuration of a stack that was defined previously.

Automatically Adding or Replacing Nodes in a Stack

You can easily automatically add or replace nodes in an existing stack.

Important
You can only replace a switch with the same switch model, or with a model from the same switch series (family) with the same number of ports, but a different port type. For example, you can replace an ExtremeSwitching 5420F-24T-4XE with an ExtremeSwitching 5420-24P-4XE (24T → 24P).

Connecting a new node to a stack automatically triggers the following tasks in a stack:

- Adds New Nodes—When a new switch is connected to a stack, the switch is automatically:
  1. Configured with the proper stacking support parameters needed to bring up either native or alternate stack ports. This automatic configuration (auto-discovery) is supported on 5420 and 5520 switches. All other Universal switches require that you configure stacking support on the stacking ports manually.
  2. Assigned the lowest available slot number.
  3. Set as stacking master-capability if the stack has only one master-capable switch (the master itself) and the switch is directly connected to the current stack master. Otherwise, the new switch is not made master-capable.
  4. Assigned the stack MAC address.
  5. Rebooted, which then causes the new node to join the stack.
- Incompatible ExtremeXOS Check—Checks for and resolves incompatible ExtremeXOS software versions of stack nodes. If the new switch is running an incompatible version, the version is reconciled by running the `synchronize {slot slotid} {file_name}` command for that switch causing the new switch to align with the stack master.

  Note
  You cannot automatically update a switch running ExtremeXOS 30.2 or earlier to ExtremeXOS 30.3 or later due to a file system compatibility issue. If a switch has ExtremeXOS 30.2 or earlier, prior to inserting the switch into the stack topology, you need to manually upgrade the switch (see Upgrading the Software on all Active Nodes on page 191).

- **License Mismatch Check**— Checks for and resolves software license mismatches between all master-capable nodes. The following command is issued: `configure stacking {node-address node-address | slot slot-number} license-level license_restriction`.

  Note
  License mismatches due to installing feature licenses cannot be resolved automatically. The node is left in the failed state.

To start automatic node replacement:

1. Prepare the replacement node:
   a. If the replacement switch is not out of the shipping box, remove any configuration and restore the factory defaults by running the command `unconfigure switch {all | erase [all | nvram]}` with the all option.
   b. If the replacement switch is going to be the backup node, check for feature license mismatches between it and the master by running `show licenses {[slot slot |all]} {detail}` on both switches. Install licenses as needed to the replacement node by running the command `enable license {software} [key ]` or;

      `install license file filename for universal platforms.`
   c. Power down the replacement switch.
2. Remove the current node from the stack.
3. If you are changing to a different port type (for example, 24T → 24P):
   a. On the master, run the command `configure slot slot module module_type` , and change the module_type as applicable.
   b. Save the configuration (`save configuration {primary | secondary | existing-config | new-config}`).
4. Physically connect the new switch to the stack, and power it up. For information about cabling, see the *hardware installation guide* for your switch.
5. Configure the switch port and speed for stack communication on the connected port of the new node for ExtremeSwitching Universal switches by running the command configure stacking-support stack-ports.

To manually add or replace a node, see Manually Adding Nodes to a Stack on page 199.

Manually Adding Nodes to a Stack

Tip
For an easier, automatic way to add or replace a node, see Automatically Adding or Replacing Nodes in a Stack on page 198.

Adding a new switch, or node, to an active stack topology is similar to bringing up a new stack.

Note
If the node being added is actually a replacement node for one that was previously removed, see Replacing a Node with the Same Switch Type on page 202 or Replacing a Node with a Different Switch Type on page 204.

Note
If the stack will use MPLS, only the 5520 switch can act as primary and backup.

Review the general and model-specific configuration guidelines for the node you are installing. These guidelines are described in the *hardware installation guide* for your switch.

The examples in the following procedure assume that your current stack has four nodes and you are adding a new node at slot 5.

To manually add a node to a stack:

1. Before connecting the new node to the stack, prepare it as follows:
   a. Turn off stacking auto-discovery by using the following command:
      `configure` **`stacking-support auto-discovery`** `[`**`disable`** `|` **`enable`**`]`
   b. With the power off, install any required option cards as described in the *hardware installation guide* for your switch.
   c. Power on the new node.
   d. Use the `show switch` command to verify that the new node's software is compatible with the stack:
      - The new node must run the same ExtremeXOS version as the stack. Install the correct version if necessary.
      - The ExtremeXOS software must be booted on the same image (primary or secondary) as the stack. If the new node is booted on a different image, change the image before you continue.
   e. Use the `enable stacking` command to enable stacking. Then decline the Easy Setup option.
   f. Configure a unique slot number for the new node (see Configuring Slot Numbers on page 168).

      Select the lowest slot number that is not already in use in the stack. In the following example, the new node has MAC address 00:04:96:26:6c:92 and is assigned to slot 5.

      ```
      Switch.3 # configure stacking node-address 00:04:96:26:6c:92 slot-number 5
      This command will take effect at the next reboot of the specified node(s).
      ```

   g. Configure the node's primary-capability to correspond to the role it should have in the stack (see Configuring Master-Capability on page 197).

h. If the new node will operate as a primary-capable node, use the `show licenses` command to verify that the enabled license level is at the same level as the primary-capable nodes in the stack.

   If necessary, configure the license-level restriction of the new node to be same as the other primary-capable nodes in the stack (see Managing Licenses on a Stack on page 184)(Extreme Platform ONE license applies to all nodes in stack).

i. Configure the node role priority to correspond to the priority it should have in the stack (see Configuring the Primary, Backup, and Standby Roles on page 169).

j. Configure an alternate IP address and gateway (see Configuring an Alternate IP Address and Gateway on page 192).

k. If the new node will use the SummitStack-V feature, configure the alternate stacking ports as described in Use Ethernet Ports for Stacking (SummitStack-V Feature) on page 150.

2. Connect the stacking cables to the new node.

   The connections should be made such that the new node appears in the natural position in the stack and in the slot. The following example adds a new node that becomes slot 5.

   - Break the connection between slot 4 port 2 and slot 1 port 1.
   - Connect slot 5 port 1 (the new node) to slot 4 port 2.
   - Connect slot 5 port 2 (the new node) to slot 1 port 1.

   For more information about cabling, see the *hardware installation guide* for your switch.

3. Reboot the new node.
4. At the stack primary node, enter the command `synchronize stacking node-address` *`node-address`*.
5. Reboot the new node by entering the command `reboot node-address` *`node-address`*`]`.
6. (Optional) Run the `show stacking` and `show slot` commands, as shown in the following example, to verify that the configuration is what you want.

```
Slot-1 Stack.1 # show stacking
Stack Topology is a Ring
Active Topology is a Ring
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:9c:e4:39  1     Active       Master   CA-
00:04:96:9b:c1:34   2     Active       Backup   CA-
00:04:96:9e:5c:76   3     Active       Standby  CA-
00:04:96:9c:53:b6   4     Active       Standby  CA-
00:04:96:26:6c:92   5     Active       Standby  CA-
* - Indicates this node
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology

Slot-1 Stack.2 # show slot
Slots    Type                Configured          State       Ports
-------------------------------------------------------------------
Slot-1   5420-24t-4xe                            Operational   28
Slot-2   5420-24p-4xe                            Operational   28
Slot-3                                           Empty          0
Slot-4                                           Empty          0
```

```
Slot-5                                          Empty           0
Slot-6                                          Empty           0
```

Replacing a Node with the Same Switch Type

Tip
For an easier, automatic way to add/replace a node, see Automatically Adding or Replacing Nodes in a Stack on page 198.

When you replace a node with the same switch type, for example when you replace a 5420-24T-4XE switch with another5420-24T-4XE switch, you can continue to use the same stack configuration.

If you are replacing a node with a different switch type, you must change the stack configuration before the new node can operate. Follow the procedure in Replacing a Node with a Different Switch Type on page 204.

Note
If the stack will use *MPLS (Multiprotocol Label Switching)*, only the 5520 can act as primary and backup.

To replace a node with an identical switch type, follow these steps:

1. Use the `show switch`, `show licenses`, and `show stacking configuration` commands to display configuration information for the node that is being replaced.

   Note the following attributes of the node you are replacing:

   - ExtremeXOS software version
   - Partition on which the switch is booted
   - Effective license level for the stack
   - Slot number
   - primary-capable feature configuration
   - Node priority
   - Alternate gateway IP address
2. Remove the stacking cables from the node that is being replaced.
3. Replace the node with another switch of the same type.
4. Before connecting the new (replacement) switch to the stack, prepare the switch as follows:
   a. Review the attributes needed for the node you are installing, as listed in step 1.
   b. With the power off, install any required option cards as described in the *hardware installation guide* for your switch.
   c. Power on the new node.

d. Use the `show switch` command to verify that the new node's software is compatible with the stack:
   - The new node must run the same ExtremeXOS version as the stack. Install the correct version if necessary.
   - The ExtremeXOS software must be booted on the same image (primary or secondary) as the stack. If the new node is booted on a different image, change the image before you continue.

e. Use the `enable stacking` command to enable stacking. Then decline the Easy Setup option.

f. Configure the slot number for the new node using the slot number noted in step 1. For more information, see Configuring Slot Numbers on page 168.

g. Configure the node's primary-capability to correspond to the role it should have in the stack (see Configuring Master-Capability on page 197).

h. If the new node will operate as a primary-capable node, use the `show licenses` command to verify that the enabled license level is at the same level as the primary-capable nodes in the stack.

   If necessary, configure the license-level restriction of the new node to be same as the other primary-capable nodes in the stack (see Managing Licenses on a Stack on page 184)(Extreme Platform ONE license applies to all nodes in stack).

i. Configure the node role priority to correspond to the priority it should have in the stack (see Configuring the Primary, Backup, and Standby Roles on page 169).

j. Configure an alternate IP address and gateway (see Configuring an Alternate IP Address and Gateway on page 192).

k. If the new node will use the SummitStack-V feature, configure the alternate stacking ports as described in Use Ethernet Ports for Stacking (SummitStack-V Feature) on page 150.

5. Connect the stacking cables and reboot the node. The new node will join the stack topology.

   For cabling instructions, see the *hardware installation guide* for your switch.

6. At the stack primary node, enter `synchronize stacking`.

   Note
   If the primary node was replaced, log into another stack node before entering this command.

7. Reboot the new node by entering the command: `reboot slot [`*`slot-number`* `|` **`node-address`** *`node-address`*`]`.

   Note
   If the primary node was replaced, reboot the stack by entering the reboot command at the primary node.

8. (Optional) Run the `show stacking configuration` command and verify that the configuration is what you want.

   Note
   To verify that the new node became operational, enter the `show slot {slot {detail} | detail }` command. If the slot shows a Mismatch state, the node was replaced with a different type of switch (see Replacing a Node with a Different Switch Type on page 204).

Replacing a Node with a Different Switch Type

Tip
For an easier, automatic way to add/replace a node, see Automatically Adding or Replacing Nodes in a Stack on page 198.

When you replace a node with a different switch type, you cannot continue to use the same stack configuration. The slot configuration for the replaced node must change to reflect the new switch type.

Note
If you are replacing a node with the same switch type, you can continue to use the existing stack configuration. For more information, see Replacing a Node with the Same Switch Type on page 202.

To replace a node with a different switch type, follow these steps:

1. Use the `show switch`, `show licenses`, and `show stacking configuration` commands to display configuration information for the node to be replaced.

   Note the following about the node you are replacing:
   - ExtremeXOS software version
   - Partition on which the switch is booted
   - Effective license level for the stack
   - Slot number
   - Master-capable feature configuration
   - Node priority
   - Alternate gateway IP address
2. Enter the `unconfigure slot slot` command to remove the configuration for the node to be replaced.

   All configuration parameters (except for the related node's NVRAM-based configurations such as stacking parameters, image to be used, and failsafe account) for the slot are erased.
3. Remove the stacking cables from the node that is being replaced.
4. Add the new node to the stack following the procedure in Manually Adding Nodes to a Stack on page 199.

Merging Two Stacks

You can join or merge two stacks to create one larger stack. However, the maximum number of nodes in an active topology is eight.

The operation performed when two stack segments are joined together depends on the following factors:

- Whether a slot number is duplicated
- Whether both stacks have master nodes
- The states of the nodes in each stack

If the nodes are configured with stacking enabled, one of the following occurs:

- If two segments are joined, both have operational masters, and at least one of the nodes in one of the stacks duplicates a slot number of a node in the other stack, the join is allowed. The link that has just connected the two stacks shows as Inhibited. This prevents accidental stack joins. In this condition, the nodes on the joined segment can still be reconfigured centrally for stacking.
- If two segments are joined, both have operational masters, and all nodes have assigned slot numbers that are unique in both stacks, the dual master situation is automatically resolved.
- If two segments are joined, there are no duplicate slot numbers, one of the segments has a master and a backup node, and the other segment does not have either a master or a backup node, the nodes in this segment are acquired by the master node. These nodes become standby nodes in the stack.

The nodes that are not configured for stacking do not attempt to join the active topology but join the stack anyway.

Any nodes enabled for stacking that are isolated between nodes (that are not enabled for stacking) attempt to form an isolated active topology.

If one of the nodes that is not configured for stacking is then configured for stacking and restarted, the behavior is as if two active stacks were joined.

Example: Merging Two Stacks

This example demonstrates how to join two stacks, named StackA and StackB. Each stack contains two switches (nodes), and the joined stack contains four nodes.

In the steps that follow, the nodes are referred to as A1, A2, B1, and B2, as shown in this table:

Table 21: Example Node Names in Stacks Being Merged

Node Name	Original Stack	Original Slot	MAC Address
A1	StackA	1	00:04:96:9c:e4:39
A2	StackA	2	00:04:96:9b:c1:34
B1	StackB	1	00:04:96:9e:5c:76
B2	StackB	2	00:04:96:9c:53:b6

Here are displays taken from the original StackA:

```
Slot-1 Stack.1 # show stacking
Stack Topology is a Ring
Active Topology is a Ring
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:9c:e4:39  1     Active       Master   CA-
 00:04:96:9b:c1:34  2     Active       Backup   CA-
* - Indicates this node
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology

Slot-1 Stack.2 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured

Slot-1 Stack.3 # show stacking stack-ports
Stack Topology is a Ring
Slot Port Select Node MAC Address  Port State  Flags Speed
---- ---- ------ ----------------- ----------- ----- -----
*1   1    27     00:04:96:9c:e4:39 Operational C-      10G
*1   2    28     00:04:96:9c:e4:39 Operational CB      10G
 2   1    27     00:04:96:9b:c1:34 Operational CB      10G
 2   2    28     00:04:96:9b:c1:34 Operational C-      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is Blocked

Slot-1 Stack.4 # show slot
Slots    Type                Configured          State       Ports
-------------------------------------------------------------------
Slot-1   switch_model                            Operational   28
Slot-2   switch_model                            Operational   28
Slot-3                                           Empty          0
Slot-4                                           Empty          0
Slot-5                                           Empty          0
Slot-6                                           Empty          0
Slot-7                                           Empty          0
Slot-8                                           Empty          0
```

Here are displays taken from StackB:

```
Slot-1 Stack.1 # show stacking
Stack Topology is a Ring
Active Topology is a Ring
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:9e:5c:76  1     Active       Master   CA-
 00:04:96:9c:53:b6  2     Active       Backup   CA-
* - Indicates this node
```

```
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology

Slot-1 Stack.2 # show stacking configuration
Stack MAC in use: 02:04:96:9e:5c:76
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9e:5c:76 1   1   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9c:53:b6 2   2   Auto <none>             <none>          CcEeMm-Nn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured

Slot-1 Stack.3 # show stacking stack-ports
Stack Topology is a Ring
Slot Port Select Node MAC Address  Port State  Flags Speed
---- ---- ------ ----------------- ----------- ----- -----
*1   1    15     00:04:96:9e:5c:76 Operational C-      10G
*1   2    16     00:04:96:9e:5c:76 Operational CB      10G
 2   1    15     00:04:96:9c:53:b6 Operational CB      10G
 2   2    16     00:04:96:9c:53:b6 Operational C-      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is Blocked

Slot-1 Stack.4 # show slot
Slots    Type                Configured          State       Ports
-------------------------------------------------------------------
Slot-1   switch_model                            Operational   16
Slot-2   switch_model                            Operational   16
Slot-3                                           Empty          0
Slot-4                                           Empty          0
Slot-5                                           Empty          0
Slot-6                                           Empty          0
Slot-7                                           Empty          0
Slot-8                                           Empty          0
```

1. Form the new stack. Assuming both stacks are rings, break one link in each stack as follows:

   Note
   Refer to Table 21 for a guide to the switch (node) names used in the following steps.

   a. For StackA, break the link between node A1 port 1 and node A2 port 2.
   b. For StackB, break the link between node B1 port 1 and node B2 port 2.
2. Connect the broken links between the two stacks to form a ring as follows:
   a. Connect node A2 port 2 to node B1 port 1.
   b. Connect node B2 port 2 to node A1 port 1.
      Because both stacks are active stacks and have duplicate slot numbers, the links between the two stacks are in Inhibited state.
3. Assume that the master node of stackA is to be the master node of the joined stack. Log into the intended master node. (This is node A1.)

4. Verify the details of the new stack using the following commands:
   `show stacking`
   `show stacking configuration`
   `show stacking stack-ports`

```
Slot-1 Stack.1 # show stacking
Stack Topology is a Ring
Active Topology is a Daisy-Chain
Node MAC Address    Slot  Stack State  Role     Flags
------------------  ----  -----------  -------  ---
*00:04:96:9c:e4:39  1     Active       Master   CA-
 00:04:96:9b:c1:34  2     Active       Backup   CA-
 00:04:96:9e:5c:76  1     Active       Master   --O
 00:04:96:9c:53:b6  2     Active       Backup   --O
* - Indicates this node
Flags:  (C) Candidate for this active topology, (A) Active Node
        (O) node may be in Other active topology

Slot-1 Stack.2 # show stacking configuration
Stack MAC in use: 02:04:96:9c:e4:39
Node               Slot         Alternate          Alternate
MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
------------------ --- --- ---- ------------------ --------------- --------- ---
*00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
 00:04:96:9e:5c:76 1   1   Auto <none>             <none>          CcEe--iNn --
 00:04:96:9c:53:b6 2   2   Auto <none>             <none>          CcEe--iNn --
* - Indicates this node
Flags:  (C) master-Capable in use, (c) master-capable is configured,
        (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
        (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
        (i) Stack MACs configured and in use are not the same or unknown,
        (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
        (-) Not in use or not configured
License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
        (c) Core, (a) Advanced edge, or (e) Edge configured,
        (-) Not in use or not configured

Slot-1 Stack.3 # show stacking stack-ports
Stack Topology is a Ring
Slot Port Select Node MAC Address  Port State  Flags Speed
---- ---- ------ ----------------- ----------- ----- -----
*1   1    27     00:04:96:9c:e4:39 Inhibited   --      10G
*1   2    28     00:04:96:9c:e4:39 Operational C-      10G
 2   1    27     00:04:96:9b:c1:34 Operational C-      10G
 2   2    28     00:04:96:9b:c1:34 Inhibited   --      10G
 1   1    15     00:04:96:9e:5c:76 Inhibited   --      10G
 1   2    16     00:04:96:9e:5c:76 Operational C-      10G
 2   1    15     00:04:96:9c:53:b6 Operational C-      10G
 2   2    16     00:04:96:9c:53:b6 Inhibited   --      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is
Blocked                                          Empty          0
```

5. Configure the nodes so that they all have unique slot numbers.
   See Configuring Slot Numbers on page 168.
6. Configure the stack MAC address using the `configure stacking mac-address` command.
   See Assigning a MAC Address to a Stack on page 172.
7. Optionally, use the `configure stacking redundancy minimal` command to ensure that only slots 1 and 2 are master-capable.

8. Optionally, configure new alternate IP addresses for nodes from the original StackB.

   See Configuring an Alternate IP Address and Gateway on page 192.

9. For master-capable nodes, configure a license restriction to be the minimum of the two original values on all master-capable nodes.

   Alternatively, you may purchase license upgrades from Extreme if necessary. In this case, use the command: `configure stacking license-level edge` (not required with an Extreme Platform ONE license).

10. Either reboot the entire stack topology using the `reboot stack-topology` command, or individually reboot the two nodes that formerly were in StackB.

    Using the nodes in this example, the latter requires the following commands:

    ```
    reboot node 00:04:96:9e:5c:76
    reboot node 00:04:96:9c:53:b6
    ```

    Reboot the nodes in this order: standby nodes first, backup node next, and master node last. Because none of these nodes is master-capable, there is no temporary dual master situation as a result of these separate node reboots.

11. When the rebooted nodes come back up, run the following commands to see the resulting stack.

    You can verify that the joined stack came up as expected – that is, all nodes have unique slot numbers, a common stack MAC address, and so forth:

    ```
    Slot-1 Stack.15 # show stacking
    Stack Topology is a Ring
    Active Topology is a Ring
    Node MAC Address    Slot  Stack State  Role     Flags
    ------------------  ----  -----------  -------  ---
    *00:04:96:9c:e4:39  1     Active       Master   CA-
     00:04:96:9b:c1:34  2     Active       Backup   CA-
     00:04:96:9e:5c:76  3     Active       Standby  CA-
     00:04:96:9c:53:b6  4     Active       Standby  CA-
    * - Indicates this node
    Flags:  (C) Candidate for this active topology, (A) Active Node
            (O) node may be in Other active topology

    Slot-1 Stack.16 # show stacking configuration
    Stack MAC in use: 02:04:96:9c:e4:39
    Node               Slot         Alternate          Alternate
    MAC Address        Cfg Cur Prio Mgmt IP / Mask     Gateway         Flags     Lic
    ------------------ --- --- ---- ------------------ --------------- --------- ---
    *00:04:96:9c:e4:39 1   1   Auto <none>             <none>          CcEeMm-Nn --
     00:04:96:9b:c1:34 2   2   Auto <none>             <none>          CcEeMm-Nn --
     00:04:96:9e:5c:76 3   3   Auto <none>             <none>          CcEeMm-Nn --
     00:04:96:9c:53:b6 4   4   Auto <none>             <none>          CcEeMm-Nn --
    * - Indicates this node
    Flags:  (C) master-Capable in use, (c) master-capable is configured,
            (E) Stacking is currently Enabled, (e) Stacking is configured Enabled,
            (M) Stack MAC in use, (m) Stack MACs configured and in use are the same,
            (i) Stack MACs configured and in use are not the same or unknown,
            (N) Enhanced protocol is in use, (n) Enhanced protocol is configured,
            (-) Not in use or not configured
    License level restrictions: (C) Core, (A) Advanced edge, or (E) Edge in use,
            (c) Core, (a) Advanced edge, or (e) Edge configured,
            (-) Not in use or not configured

    Slot-1 Stack.17 # show stacking stack-ports
    Stack Topology is a Ring
    Slot Port Select Node MAC Address  Port State  Flags Speed
    ---- ---- ------ ----------------- ----------- ----- -----
    ```

```
*1   1    27     00:04:96:9c:e4:39 Operational C-      10G
*1   2    28     00:04:96:9c:e4:39 Operational CB      10G
 2   1    27     00:04:96:9b:c1:34 Operational CB      10G
 2   2    28     00:04:96:9b:c1:34 Operational C-      10G
 3   1    15     00:04:96:9e:5c:76 Operational C-      10G
 3   2    16     00:04:96:9e:5c:76 Operational C-      10G
 4   1    15     00:04:96:9c:53:b6 Operational C-      10G
 4   2    16     00:04:96:9c:53:b6 Operational C-      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is Blocked

Slot-1 Stack.18 # show slot
Slots    Type                 Configured           State       Ports
-------------------------------------------------------------------
Slot-1   switch_model                              Operational   28
Slot-2   switch_model                              Operational   28
Slot-3   switch_model                              Operational   16
Slot-4   switch_model                              Operational   16
Slot-5                                             Empty          0
Slot-6                                             Empty          0
Slot-7                                             Empty          0
Slot-8                                             Empty          0
```

12. Configure the new slots in VLANs, IP subnetworks, and so forth as required.

Removing a Node from a Stack

To remove a node from a stack, follow these steps:

1. Determine if the target node to be removed is using the SummitStack-V feature by issuing the `show stacking stack-ports` command.

   Examine the Select column to see whether the target node is using alternate (non-native) stacking ports.
2. If the target node is using alternate stacking ports, do the following:
   a. Log into the target node and issue the command: `unconfigure stacking-support`
   b. Log out of the target node.

   Note
   Do not reboot the target node at this time.
3. Log into the master node.
4. Delete the target node stacking configuration by entering the following command:

   `unconfigure stacking {`**`node-address`** *`node_address`* | **`slot`** *`slot_number`*`}`
5. Reboot the target node by entering the following command:

   `reboot [`**`node`** *`node-address`* | **`slot`** *`slot-number`*`]`

   When the node reboots, it detects that the configuration file selected is a stacking configuration file (see Stack Configuration Parameters, Configuration Files, and Port Numbering on page 160). It de-selects the configuration file and uses the factory defaults.
6. Disconnect the node from the stack, and redeploy it as needed.

Dismantling a Stack

To dismantle a stack and use the ExtremeSwitching switches in standalone mode, do the following:

1. Issue the `show stacking stack-ports` command to see if any nodes in the stack are using alternate stacking ports.

   In the following example, the switches in slots 1 and 2 have 24 data ports, and the switches in slots 3 and 4 have 12 data ports. The numbers in the **Select** column show that all ports in this stack are alternate stacking ports, not regular data ports.

```
Slot-1 Stack.5 # show stacking stack-ports
Stack Topology is a Ring
Slot Port Select Node MAC Address  Port State  Flags Speed
---- ---- ------ ----------------- ----------- ----- -----
*1   1    27     00:04:96:9c:e4:39 Operational C-      10G
*1   2    28     00:04:96:9c:e4:39 Operational CB      10G
2   1    27     00:04:96:9b:c1:34 Operational CB      10G
2   2    28     00:04:96:9b:c1:34 Operational C-      10G
3   1    15     00:04:96:9e:5c:76 Operational C-      10G
3   2    16     00:04:96:9e:5c:76 Operational C-      10G
4   1    15     00:04:96:9c:53:b6 Operational C-      10G
4   2    16     00:04:96:9c:53:b6 Operational C-      10G
* - Indicates this node
Flags:  (C) Control path is active, (B) Port is Blocked
```

2. For every non-master node in the stack that is using alternate stacking ports, log into the node and issue the `unconfigure stacking-support` command.

   Do not reboot any of the switches.

   If step 1 showed that no alternate stacking ports are in use, skip this step.

   Note

   - This command resets the ports so that they are no longer dedicated for use as stacking ports.
   - It is not necessary to unconfigure stacking-support on the master node.
   - If a node is a member of the active topology, you can log in to other nodes using the `telnet slot` *`slot-number`* {**`no-auto-login`**} command. Otherwise, you need access to the node's console port, or you need to log in through a management network.

3. Log into the master node and issue the `unconfigure switch all` command.

   After this command is entered, the configuration file is deselected, all stacking parameters are reset to factory defaults, and all nodes in the active topology reboot. In effect, this sets all nodes back to the factory default configuration, thus allowing each switch to be redeployed individually.
4. Restore the saved configuration for each switch, if applicable.

Troubleshooting a Stack

Use this section to diagnose and troubleshoot problems with your stacked switches.

The `show stacking`, `show stacking configuration`, `show stacking-support`, and `show stacking stack-ports` commands can help you identify switches and stack ports that are improperly configured, not properly cabled, or powered down.

The commands can help you spot common problems like the following.

Incorrect ExtremeXOSSoftware Version

If a node appears in the stack as expected but does not appear to be operating as configured, use the `show slot {`*`slot`* `{`**`detail`**`} | `**`detail`**` }` command to see if there is a license mismatch or if the node is running an incorrect ExtremeXOS software version. For more information, see Managing Licenses on a Stack on page 184.

Stacking Not Enabled

If the `show stacking` command displays the status as Disabled for any node in the stack, use the `enable stacking {`**`node-address`** *`node-address`*`}` command, with *`node-address`* as the disabled node address, to enable stacking on that node. You can issue the command from the master node, and you can reboot the disabled node from the master node to activate the slot number configuration.

Choice of Master Node

If the switch with the highest priority was not elected master, it might be because the stack nodes were powered up at different times. Reboot all nodes in the stack simultaneously.

The following topics contain information for troubleshooting problems related to the choice of the master node:

- Managing a Dual Primary Situation on page 213
- Connecting to a Stack with No Primary on page 216
- Rescuing a Stack that has No Primary-Capable Node on page 216

Choice of Backup and Standby Nodes

About five minutes after a master node takes control of the stack, you might see one of the following messages:

```
Warning: The Backup stack node is not as powerful or as capable
as the Master stack node. This configuration is not recommended
for successful use of the failover feature.
Notice: There are Standby stack nodes which are more powerful and more capable
than the Master and/or Backup stack nodes. This configuration is not recommended
for optimal stack performance. We recommend that you reconfigure the stacking
master-capability and/or priority parameters to allow the higher performing and
more capable nodes to become Master and/or Backup stack nodes.
```

In each case, to optimize your use of the failover feature, follow the guidelines in Configuring the Primary, Backup, and Standby Roles on page 169.

Loss of Saved Files

Note that saved files (backup configurations, script, etc.) are lost on a node that becomes a non-master when implementing stacking. Disabling or deleting the stacking configuration does not restore the files.

Refer to the following topics for help with troubleshooting other problems in your stack.

Managing a Dual Primary Situation

If a daisy chain is broken, or if a ring is broken in two places, it is possible to form two separate active stack topologies. This results in a dual primary situation, as shown in the following example .

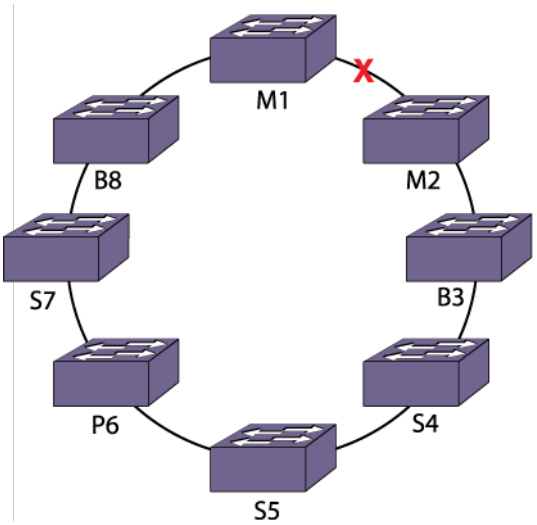


Figure 6: Example of a Split Stack that Results in a Dual Primary Situation

P6	Node 6 is powered off
M	Primary nodes
B	Backup nodes
S	Standby nodes
X	Indicates the broken link

In the example, a link was broken while a node in the ring was powered off. The broken link formerly connected the original primary (M1) and backup (M2) nodes of a single active topology.

All nodes in the stack except the powered-off node are in the active topology and all nodes are configured to be primary-capable. Nodes 1, 7 and 8 form an active topology and nodes 2, 3, 4, and 5 form another active topology. Node M2 immediately transitions from backup to primary node role. Nodes B8 and B3 are elected in their respective active topologies as backup nodes.

If the backup node is on one stack and the primary node is on the other, the backup node becomes a primary node because the situation is similar to that of primary failure.

Because both stacks are configured to operate as a single stack, there is confusion in your networks. For example, all of the switch's configured IP addresses appear to be duplicated. The management IP address also appears to be duplicated since that address applies to the entire original stack.

To help mitigate the dual primary problem, you can configure primary-capability so as to prevent some nodes in the stack from operating in backup or primary node roles. In addition, you can force all nodes in the (broken) stack topology to restart and come up as not primary-capable for the life of that restart. The `save configuration {primary | secondary | existing-config | new-config}` command saves the configuration on all nodes in the active topology.

Standby nodes that exist in a severed stack segment that does not contain either the original primary or backup node do not attempt to become the primary node. Instead, these nodes reboot. After rebooting, however, a primary election process occurs among the nodes on this broken segment, resulting in a dual primary situation.

Dual primary conditions are also possible when two non-adjacent nodes in a ring or a single (middle) node in a daisy chain reboot.

For a period of time, a rebooting node does not advertise itself to its neighbors, resulting in temporary stacking link failures. This can cause node isolation, and the nodes that are isolated perform as a severed stack segment depending on the circumstances of the severance:

- If the backup node is on the broken portion, it becomes a (dual) primary.
- If the backup node is on the same portion as the primary, all nodes on the (other) broken portion reboot.

When the rebooting nodes have sufficiently recovered, or when a severed stack is rejoined, the dual primary condition is resolved, resulting in the reboot of one of the primary nodes. All standby and backup nodes that had been acquired by the losing primary node also reboot.

You can avoid a dual primary possibility during configuration by:

- Configuring the stack in a ring topology.
- Avoiding having too many primary-capable nodes when configuring larger stacks.
- Placing the primary-capable nodes that provide stack redundancy in such a way that broken stacking links are unlikely.

Eliminating a Dual Master Situation Manually

To eliminate the dual master situation, you need to know all the nodes that are supposed to be in the stack. You might lose the management connectivity to

the master node because the other master node duplicates the stack's primary management IP address and stack MAC address.

Note
The following procedure is necessary only if you cannot reconnect the severed link in a timely manner. If you can reconnect, the dual master condition resolves itself. The formerly broken portion of the stack reboots and the nodes come up as standby nodes.

1. If you lose the management connectivity, log into the master node using its alternate management IP address.
2. Use the `show stacking` command to determine which nodes have been lost from the stack.

   You should already know which nodes are expected to be part of the stack.
3. Log into any node in the severed segment you want to deactivate, either through its console port or through the management interface using the alternate management IP address.
4. Issue the `show stacking` command to find out whether the broken segment has indeed elected a new master node.
5. Using the `reboot stack-topology as-standby` command, reboot the broken segment.

   This forces all nodes in the segment to come up as standby nodes.

   If you have unsaved configuration changes, take care when selecting the stack segment to be rebooted. You should reboot the segment that has the smaller System UpTime.

   If you know the node that was master of the unbroken stack, you can reboot the stack segment that does not contain this master node. Otherwise, determine the System UpTime shown by each master node.

   If the System UpTimes of both masters are the same, you can reboot either segment without loss of unsaved configuration changes. If the System UpTimes of both masters differ, you must reboot the segment with the smaller System UpTime.

Automatic Resolution of the Dual Master Situation

When two stack segments are connected together and no slot number is duplicated on either segment, it is assumed that this is a severed stack rejoin.

It is possible that each stack segment has its own master. Resolution of the dual master situation should generally be in favor of the original stack segment's master node. This is because the original stack segment may still retain the unsaved configuration. If the severed segment was restarted before electing a new master node, the unsaved configuration is lost on that segment.

The master election is done using the System UpTime. The master election process collects the System UpTime information of the nodes. If a failover occurs, the System UpTime is inherited by the new master node, and the new master node continues to increase it as time passes. Thus the System UpTime is the time since a master was first

elected on a segment. When the stack is broken and both master and backup nodes are on the same segment, the severed segment always has the smaller System UpTime.

If a stack severance results in the master and backup nodes being on different segments, both have the same System UpTime. In this case, the master is elected using the normal node role election method.

Connecting to a Stack with No Primary

If an entire stack has no primary node because the stack has been rebooted in standby only mode, you can log in to a node by using the failsafe account.

If a new node has been added to the stack since the stack failsafe account was configured, logging in to that node requires knowledge of the failsafe account information that is already configured into that node's NVRAM.

If you do not know the failsafe account and you still want to log in to the stack, you have to:

1. Join the stack to another segment that has a primary node to which the you have access.
2. Manually restart the stack to clear the as-standby condition if the reboot stack-topology as-standby command was previously used.
3. Use the procedure described in Rescuing a Stack that has No Primary-Capable Node on page 216.

Rescuing a Stack that has No Primary-Capable Node

It is possible for all nodes in a stack to have primary-capability set to OFF. For example, if a stack was operating with no redundancy (for example, with one primary-capable node) and the primary node failed, all other nodes in the stack restart as standby nodes and there is no primary node.

Another example is the case where you dismantle a stack before using the `unconfigure stacking` command or the `unconfigure switch all` command. In this case, the individual switches are configured for stacking, are not primary-capable, and are isolated from a stack primary.

In this situation, the only security information available is the failsafe account. If you know the failsafe user name and password, you can log into any node and reconfigure primary-capability or redundancy. However, if you do not know the failsafe account information, there is another way you can change the configuration.

The procedure described here generally is not needed if another primary-capable node is expected to rejoin the stack. If this procedure is used, it is possible that the new primary will duplicate the primary that is expected to rejoin later.

To assign a new primary-capable node, follow these steps.

1. At the login prompt, enter the following special login ID exactly as displayed below (all uppercase letters) and press **[Enter]**:

   ```
   REBOOT AS PRIMARY-CAPABLE
   ```

   The following message appears:

   ```
   Node reboot initiated with primary-capability turned on.
   ```

   This node then sets an internal indicator that is preserved across the reboot. While restarting, the node notices and resets this indicator, ignores the node primary-capability configuration, and becomes a primary node.

   The special login ID described here is available only if all of the following conditions are met:

   - The node supports the SummitStack feature.
   - Stacking mode is active on the node.
   - All nodes in the active topology have primary-capability turned off.
   - There is no primary node in the active topology.

   If the above conditions are met, five minutes after starting the node and every five minutes after that, the following message displays on the console:

   ```
   Warning: the stack has no Primary node and all active nodes are operating
   with primary-capability turned off. If you wish to reconfigure, you may log
   in using the failsafe account. Alternatively, you may use the special login
   REBOOT AS PRIMARY-CAPABLE with no password to force a reboot of a node with
   primary-capability temporarily turned on.
   ```

   Using the special login ID does not alter the primary-capability configuration permanently. If you restart a node that has been restarted with the special login ID, that node restarts using its configured primary-capability, unless you again use the special login ID to restart.

   When a node has been rebooted using the special login ID, it becomes a primary node. While the node is a primary, the special login ID is not recognized, even though the entire stack is still configured as not primary-capable. To get the special login ID to be recognized, the node must be rebooted again.

2. If a node was intentionally separated from the stack without first being unconfigured, its security configuration might be unusable. In this case, perform the following steps:

   a. Connect to the node's console port.
   b. Reboot the node using the special `REBOOT AS PRIMARY-CAPABLE` login ID described in step 1.
   c. During the reboot, enter the bootrom program by waiting until you see the message `Starting Default Bootloader ...` and then pressing and holding the space bar until the bootrom prompt displays.
   d. Force the switch to boot up with a default configuration by entering the following commands at the bootrom prompt:

      ```
      config none
      boot
      ```

The switch boots up in stacking mode operating as a primary-capable switch. You can then log in using the default admin account with no password.

Daisy Chain

If you issue the **show stacking** command and see that the stack topology is a daisy chain, we strongly recommend that you reconfigure the stack to use a ring topology.

Here is a sample CLI output from **show stacking** showing that the stack is in a daisy chain.

```
 # show stacking
Stack Topology is a Daisy-Chain
This node is not in an Active Topology
Node MAC Address     Slot  Stack State  Role     Flags
------------------   ----  -----------  -------   ---
*00:04:96:7d:fc:9a    -    Disabled      Master   ---
```

To solve a daisy-chain problem, follow these steps:

1. Ensure that all stack cables are properly seated and that the Link LED is illuminated on each stack port.
2. Ensure that each stack node meets the following conditions:
3. Wait for all stack nodes to finish booting and get through AAA login.
4. Enter `show stacking` to verify that all stack nodes are connected in a ring topology.
5. Check that the stacking port LEDs show as active on all nodes.
6. Reconnect the stacking cables.

Broken Stack (Isolated Nodes)

When a stack setup is not properly unconfigured before the stacking cables are removed, former stack nodes can become isolated.

A common symptom of this problem is that either or both ends of a stacking link show that the state is No Neighbor. This can mean that the port at either end is configured incorrectly. Some configuration errors can produce the No Neighbor state at one end and the Link Down state at the other end.

Here is an example of how an isolated node would look when you log in to it:

```
 (pending-AAA) login:
Warning: the stack has no Master node and all active nodes are operating with
master-capability turned off. If you wish to reconfigure, you may log in using
the failsafe account. Alternatively, you may use the special login
REBOOT AS MASTER-CAPABLE
with no password to force a reboot of a node with master-capability temporarily
turned on.
```

If the `show stacking` command shows the stack state for a slot as Active with the “O” flag set, the node associated with that slot might be isolated from the rest of the stack by a failed node.

To fix an isolated node problem, follow these steps:

1. Verify the physical connections between each port and the rest of the stack.
2. Following the prompt in the sample console session, log in by entering `reboot as master-capable`

   The switch reboots as a master-capable switch.
3. Enter `unconfigure stacking` to unconfigure the stack.

   This command takes effect only once. If the switch is powered off or rebooted you must reissue the **`reboot as master-capable`** command.

The switch is now available for use in a new stack.

Failed Stack

Even if the **`enable stacking`** command executes without any warnings or prompts, it is still possible for a stack to fail.

When a stack fails, verify that all of the following conditions are met.

A common sign of a failed stack is one of the non-master nodes being stuck at `(pending-AAA)` for more than five minutes. You can attempt to communicate with the master node (the switch on which the stack was enabled) and determine what went wrong by checking the state of each slot in the stack, as shown in this example CLI session:

```
Slot-1 Stack.1 # show slot
Slots    Type               Configured State      Ports
-------------------------------------------------------
Slot-1   switch_model       Operational            50
Slot-2   switch_model       Failed                 26
Slot-3   Empty 0
Slot-4   Empty 0
Slot-5   Empty 0
Slot-6   Empty 0
Slot-7   Empty 0
Slot-8   Empty 0
```

In this example, Slot-2 is in the failed state. To determine why it failed, enter the `show slot` command – in this example, `show slot 2` – to get detailed information about the state of the failed slot:

```
Slot-1 Stack.1 # show slot 2
Slot-2 information:
State: Failed
Download %: 0
Last Error: License Mismatch
Restart count: 1 (limit 5)
Serial number: N/A(0) N/A(0)
Hw Module Type: switch_model
Configured Type:
Ports available: 26
Recovery Mode: Reset
Node MAC: 00:00:00:00:00:00
Current State: FAIL (License Mismatch)
Image Selected:
Image Booted:
Primary ver:
```

```
Secondary ver:
Config Selected: NONE
```

In this example, the highlighted line in the output reveals the cause of the failure: a license mismatch in Slot-2.

Failed Stack Node

If the `show stacking` command shows the stack state for a slot as Failed, check the following things:

- Does the `show stacking stack-ports` command show a port state as Inhibited? If so, the problem might be a duplicate slot number. If more than one node is using the same slot number, change the slot number on one of the affected nodes to a unique slot number.
- Is the affected node isolated by other nodes for which the stack state is listed as Disabled? If so, you need to enable stacking on the disabled nodes.
- Enter the show slot detail command. If the command displays License Mismatch, either upgrade the node license, or configure a license level restriction so that all master-capable nodes are at the same effective license level.
- Enter the `show slot detail` command. If the command displays `Incompatible EXOS Version`, log into the master node and use the `synchronize slot` command to update the failed node.

License Mismatch

A stack might have a failed node even though the **`show licenses`** command shows that all nodes are running the same license and/or feature packs.

To fix a license mismatch, do the following:

1. At the master node's console, enter `show slot` *`slot_number`* repeatedly until you find the slot node that is failing.

   Note
   When you are creating a stack, a blinking orange MGMT light usually indicates a license mismatch on a master-capable standby node.

2. Enter `unconfigure stacking slot <failed_slot_number>` followed by `reboot slot <failed_slot_number>` at the master node to unconfigure the stack.
3. After making a connection to the failed switch's slot, update the license information for that node:
   a. For universal platforms, licenses can be uninstalled or revoked using the following command:

      `uninstall license [{`**`file`**`} | {`**`product`**`}] {`*`filename`*`} | {`**`productname`**`}] [{`**`withhold`**`} | {`**`revoke`**`}]`

      Select revoke.
   b. Enter `reboot`

c. Obtain the correct license information for the failed switch, then enter `# enable license` *`license_key`* (where *`license_key`* has the format *`xxxx-xxxx-xxxx-xxxx-xxxx`*).

   For more information about licensing for stack nodes, see Managing Licenses on a Stack on page 184.

4. Enter `show stacking` and `enable stacking` to re-enable the stack.

   Note
   Only those switches that are master-capable require compatible licenses be installed.

   If an Extreme Platform ONE license is applied to the stack, compatible licenses are not required.

Stacking Link Failure

A stacking link is said to be failed when one of the following happens:

- The stacking link is physically disconnected.
- The neighbor on a link stops transmitting topology information.
- The link goes down while a node restarts or when it is powered off.

Based on the stacking topology, the stack behavior changes.

Ring Topology

All traffic paths that were directed through the failed link are redirected. All nodes converge on the new (daisy chain) topology that results from the link break. The Topology Protocol that determines the stack topology immediately informs other nodes that a link has failed. Each node starts the process of redirecting traffic paths.

Daisy Chain

A stacking link failure means a severed stack. The Topology Protocol reports the loss of all nodes in the severed portion. Depending on master capability configuration and the original location of the backup node, the severed portion may or may not elect a new master node. If it does, the dual master condition may be in effect.

The `show slot {`*`slot`* `{`**`detail`**`} | `**`detail`**` }` command displays the slots that contain active nodes that are in the severed portion as Empty.

Understanding Stacking Traps

Every stack generates traps that provide status information about switches and stacking ports.

The stack routinely generates the following traps:

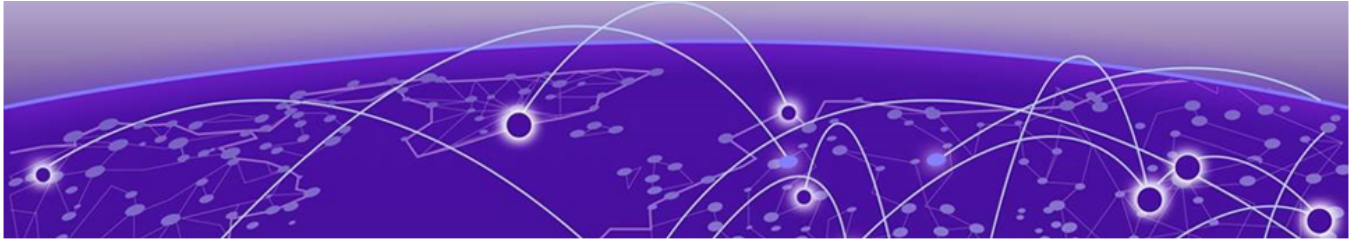
- `extremeStackMemberStatusChanged`
- `extremeStackMemberSlotId`: the slot ID
- `extremeStackMemberOperStatus`: the slot state of the switch

When an overheat condition is detected on an active node, the stack generates the following trap when the node reaches a steady state:

- `extremeStackMemberOverheat`

When a member is added to or deleted from the stack, or any time the status of a stacking port changes, the change is indicated by means of the following traps:

- `extremeStackingPortStatusChanged`
- `IfIndex`: Interface Index of the port
- `extremeStackingPortRemoteMac`: MAC Address of the remote switch attached to this port
- `extremeStackingPortLinkSpeed`: the port's link speed, for example, 100, or 1000 Mbps
- `extremeStackingPortLinkStatus`: the status of the link



Configuring Slots and Ports on a Switch

Configuring Ports on a Switch on page 223
DWDM Optics Support on page 262
Configuring Digital Diagnostic Monitoring Interface (DDMI) on page 263
Jumbo Frames on page 263
Link Aggregation on the Switch on page 265
MLAG on page 287
Mirroring on page 307
Remote Mirroring on page 312
Extreme Discovery Protocol on page 318
ExtremeXOS Cisco Discovery Protocol on page 319
Software-Controlled Redundant Port and Smart Redundancy on page 325
Port Configuration Reset to Factory Defaults on page 327
Configuring Automatic Failover for Combination Ports on page 330
Displaying Port Information on page 331
Optic Port Compatibility Check on page 331
ExtremeXOS Port Description String on page 332
Port Isolation on page 333
Energy Efficient Ethernet on page 333
Node Alias on page 335
Using Locally Administered MAC Addresses on page 339
Enabling/Disabling the USB Port on page 340

This chapter describes the processes for enabling, disabling and configuring individual and multiple ports and displaying port statistics.

Configuring Ports on a Switch

Note
A port can belong to multiple virtual routers (VRs). See Virtual Routers on page 2198 for more information on VRs.

Port Numbering

ExtremeXOS runs on stand-alone and SummitStack switches, and the port numbering scheme is slightly different on each.

Stand-alone Switch Numerical Ranges

On switches, the port number is simply noted by the physical port number.

Separate the port numbers by a dash to enter a range of contiguous numbers, and separate the numbers by a comma to enter a range of non-contiguous numbers:

- x-y—Specifies a contiguous series of ports on a stand-alone switch.
- x,y—Specifies a non-contiguous series of ports on a stand-alone switch.
- x-y,a,d—Specifies a contiguous series of ports and a non-contiguous series of ports on a stand-alone switch.
- Port:Channel—For 4120, 5120-44X-4Y-2C, 5720, 7520, and 7720 channelized ports. For example, 49:4 maps to Port 49, Channel 4.

SummitStack Numerical Ranges

On SummitStack, the port number is a combination of the slot number and the port number. The nomenclature for the port number is as follows:

```
slot:port
```

For example, if the second switch in a SummitStack has four ports, the following ports are valid:

- 2:1
- 2:2
- 2:3
- 2:4

You can also use wildcard combinations (*) to specify multiple port combinations.

The following wildcard combinations are allowed:

- slot:*: Specifies all ports on a stack node.
- slot:x-slot:y: Specifies a contiguous series of ports on multiple stack nodes.
- slot:x-y: Specifies a contiguous series of ports on a particular stack node.
- slota:x-slotb:y: Specifies a contiguous series of ports that begin on one stack node and end on another stack node.

Enabling and Disabling Switch Ports

By default, all ports are enabled. You have the flexibility to receive or not to receive *SNMP (Simple Network Management Protocol)* trap messages when a port transitions between up and down.

- To enable or disable one or more ports on a switch, use the following commands:

  `enable port [`*`port_list`* `|` **`all`**`]`

  `disable port [`*`port_list`* `|` **`all`**`]`

  For example, to disable slot 7, ports 3, 5, and 12 through 15 on a SummitStack, enter:

  ```
  disable port 7:3,7:5,7:12-7:15
  ```

- To receive these SNMP trap messages, use the following command:

  `enable snmp traps port-up-down ports [`*`port_list`* `|` **`all`**`]`

- To stop receiving these messages, use the following command:

  ```
  disable snmp traps port-up-down ports [port_list | all]
  ```

Refer to Displaying Port Information on page 331 for information on displaying link status.

Configuring Switch Port Speed and Duplex Setting

Note
For information about displaying port speed, duplex, auto-negotiation, and flow control settings, see Displaying Port Information on page 331.

ExtremeXOS supports the following port types:

- 2.5 GBASE-T ports.
- 5 GBASE-T ports.
- 10 GBASE-T ports.
- 10 Gbps ports.
- 25 Gbps.
- 40 Gbps ports.
- 50 Gbps.
- 100 Gbps ports.
- 10/100/1000 Mbps copper ports.
- 10/100/1000 SFPs.
- 10/100/1000 Mbps copper ports with *PoE (Power over Ethernet)*—only on the following switches:
  - 4220-12P-4X, 4220-24P-4X, 4220-48P-4X, 4220-4MW-20P-4X, 4220-4MW-8P-4X, and 4220-8MW-40P-4X
  - 5320-48P-8XE, 5320-24P-8XE, 5320-16P-2MXT-2X, 5320-16P-4XE, 5320-16P-4XE-DC
  - 5420F-8W-16P-4XE, 5420F-24P-4XE, 5420F-16MW-32P-4XE, 5420F-16W-32P-4XE, 5420F-48P-4XE, 5420F-48P-4XL, 5420M-24T-4YE, 5420M-24W-4YE, 5420M-16MW-32P-4YE, 5420M-24W-24S-4YE, 5420M-48T-4YE, and 5420M-48W-4YE
  - 5520-24W, 5520-48W, and 5520-12MW-36W
  - 5720 (all models)
- 10/100/1000/2500 Mbps copper ports with *PoE*—on all 4120 models
- 100/1000/2500/5000 Mbps copper ports with *PoE*—on 5720 MW models
- 100/1000/2500/5000/10000 Mbps copper ports with *PoE*—on 5720 MXW models
- 1 Gbps small form factor pluggable (SFP) fiber ports.
- 1/2.5/5 Gbps copper ports with *PoE*—only on 4220-4MW-20P-4X, 4220-4MW-8P-4X, and 4220-8MW-40P-4X
- 100 FX SFPs, which must have their speed configured to 100 Mbps.

- 10 Gbps stacking ports.
- 10 Gbps small Form Factor pluggable+ (SFP+) fiber ports. These should be configured to 10 Gbps auto off if an SFP+ optic is inserted; they should be configured to 1G auto on (or auto off) if 1G SFP optic is inserted.

Note
Stacking ports always use the same type of connector and copper PHY, which are built in to the switches. You cannot configure stacking port parameters such as port speed, duplex, and link fault signal. You also cannot configure data port features such as VLANs and link aggregation.

Auto-negotiation determines the port speed and duplex setting for each port (except 10 and 40 Gbps ports). You can manually configure the duplex setting and the speed of 10/100/1000 Mbps ports.

The 10/100/1000 Mbps ports can connect to either 10BASE-T, 100BASE-T, or 1000BASE-T networks. By default, the ports autonegotiate port speed. You can also configure each port for a particular speed (either 10 Mbps or 100 Mbps).

Note
With auto-negotiation turned off, you cannot set the speed to 1000 Mbps.

In general, SFP gigabit Ethernet ports are statically set to 1 Gbps, and their speed cannot be modified.

However, there are two SFPs supported by Extreme that can have a configured speed:

- 100 FX SFPs, which must have their speed configured to 100 Mbps.
- 100FX/1000LX SFPs, which can be configured at either speed.

The 10 Gbps ports always run at full duplex and 10 Gbps. The 40 Gbps ports always run at full duplex and 40 Gbps.

ExtremeXOS allows you to specify the medium as copper or fiber when configuring switches with combination ports. If the medium is not specified for combination ports then the configuration is applied to the current primary medium. The current primary medium is displayed in the Media Primary column of the `show ports configuration` command output.

Note
For switches that do not support half-duplex, the copper switch ports must have auto-negotiation disabled and full duplex enabled when connecting 10/100/1000 Mbps devices that do not autonegotiate. If the switch attempts and fails to auto negotiate with its partner, it will fail to link up. A non-negotiating connected device must also be manually configured for full duplex or packet loss and port errors will occur each time it detects a collision.

Note
Beginning in version 32.4, auto-negotation defaults to ON for SFP28, QSFP+, and QSFP28 ports (25G/40G/50G/100G).

To configure port speed and duplex setting , use the following command:

```
configure ports port_list {medium [copper | fiber]} auto off speed speed
duplex [half | full]
```

To configure the system to autonegotiate, use the following command:

```
configure ports port_list {medium [copper|fiber]} auto on {[{speed
speed} {duplex [half | full]}] | [{duplex [half | full]} {speed speed}]}
```

Note
The keyword medium is used to select the configuration medium for combination ports. If port_list contains any non-combination ports, the command is rejected.

ExtremeXOS does not support turning off auto-negotiation on the management port.

Support for Auto-negotiation on Various Ports

The following table lists the support for auto-negotiation, speed, and duplex setting for the various types of ports.

Port	Auto-negotiation	Speed	Duplex
100 Gbps	On (default) Off	100,000 Mbps	Full duplex
10 Gbps	Off	10,000 Mbps	Full duplex
40 Gbps	On (default) Off	40,000 Mbps	Full duplex
1 Gbps fiber SFP	On (default) Off	1,000 Mbps	Full duplex
100 FX SFP	On (default) Off	100 Mbps	Full duplex
10/100/1,000 Mbps	On (default) Off	10 and 100 Mbps 1,000 Mbps	Full/half duplex Full/half duplex
10/100 Mbps	On (default) Off	10 Mbps 100 Mbps	Full/half duplex Full/half duplex
100/1000/2.5 Gbps	On (default) Off	100 and 1,000 Mbps 2,500 Mbps	Full/half duplex Full/half duplex
100/1,000/2.5/5 Gbps	On (default) Off	100, 1000, and 2,500 Mbps 5,000 Mbps	Full duplex Full duplex
25 Gpbs	On (default) Off	25,000 Mbps	Full duplex

Port	Auto-negotiation	Speed	Duplex
10 Gbps SFP+	Off	10,000 Mbps	Full duplex
50 Gbps	On (default) Off	50,000 Mbps	Full duplex

Note
For Universal Hardware switches, auto-negotiation is disabled by default when a 1G SFP optic is inserted in a 10 Gbps SFP+ port.

Note
Half-duplex is not supported on the multi-rate ports of ExtremeSwitching 5520-12MW-36W. Half-duplex is supported on the copper ports of 48T/W and 24T/W at 10/100Mbps, but not at 1000Mbps.

Flow control on Gigabit Ethernet ports is enabled or disabled as part of auto-negotiation (see IEEE 802.3x). If auto-negotiation is set to Off on the ports, flow control is disabled. When auto-negotiation is On, flow control is enabled.

With Extreme Networks devices, the 1 Gbps ports and the 10 Gbps ports implement flow control as follows:

- 1 Gbps ports:
  - Auto-negotiation enabled
    - Advertise support for pause frames
    - Respond to pause frames
  - Auto-negotiation disabled
    - Do not advertise support for pause frames
    - Do not respond to pause frames
- 10 Gbps ports:
  - Auto-negotiation always disabled
  - Do not advertise support for pause frames
  - Respond to pause frames

Note
When 1G SFP optics are inserted into 10 GBps ports, you must configure auto-negotiation "on" with 1G speed.

Support for IEEE Forward Error Correction (FEC)

ExtremeXOS allows you to enable IEEE Forward Error Correction (FEC) Clause 91 or Clause 74 (exclusively) on a per-port basis regardless of speed/type. FEC is not supported on 10G ports.

FEC gives the receiver the ability to correct errors without requiring a reverse channel to request retransmission of data, but at the cost of a fixed, higher forward channel bandwidth. Some devices require FEC to interoperate, and certain optics use it to extend reach or improve link stability.

FEC is disabled by default on unpopulated 25G, 40G, and 100G ports. When a transceiver is inserted, FEC is enabled automatically based on the cable speed according to the following defaults:

Cable Speed	Port Speed	Default
10G	25G	Clause 91
25G	25G, 40G, 100G	Clause 91
40G	40G	Clause 74
100G	100G	Clause 91

A FEC configuration mismatch between link partners can prevent a link from coming up or cause link instability. Ensure FEC settings match on both ends of a link when configuring FEC manually.

FEC is only available on 4120 SFP28 and QSFP28 ports, 4220 SFP+ ports, 5420-4XE ports, 5420-4YE ports, 5520-VIM-4YE ports, 5720 QSFP ports, 5720-VIM-6YE/2CE ports, and Extreme 7520-48XT.

To enable/disable FEC, use the following commands:

```
configure ports port_list forward-error-correction [off | on [cl74 | cl91]]
```

To show FEC status, use the following commands:

```
show port port_list forward-error-correction {refresh | no-refresh } port-number

show port {mgmt |port_list | tag tag} information {detail}

clear counters forward-error-correction {ports {port-list | all}}
```

Displaying FEC Statistics

Version 33.7.1 adds per-port FEC operational counters to the **show ports forward-error-correction** output, providing visibility into FEC health to help diagnose marginal links, optics issues, and cabling problems.

Two counters are available for each port:

- Corrected Blocks — tracks FEC blocks or codewords that contained errors the FEC decoder successfully corrected. A non-zero value indicates a marginal but functional link.
- Uncorrectable Blocks — tracks FEC blocks or codewords that exceeded correction capability and may result in traffic loss. Any non-zero value indicates link degradation requiring investigation.

Counters are 64-bit monotonically increasing values that persist across link bounces. They reset when FEC is disabled, when the FEC mode is toggled, or when explicitly cleared. Ports that do not support FEC display blank counter values.

To display FEC statistics for all ports or a specific port list:

```
show port port_list forward-error-correction {refresh | no-refresh }
port-number
```

To clear FEC counters independently of other port counters:

```
clear counters forward-error-correction {ports {port-list | all}}
```

Note
Lane-level FEC statistics, SNMP/MIB access, and REST API/Telegraf support are not available in this release.

Configuring Extended Port Description

ExtremeXOS provides a configurable per-port “display-string” parameter that is displayed on each of the `show port` CLI commands, exposed through the *SNMP* ifAlias element, and accessible via the XML port.xsd API. This existing field is enhanced to allow up to 255 characters with much less stringent syntax limitations. Some characters are still not permitted, as they have special meanings. These characters include the following: <, >, ?, &. This new field is accessible through the `show port info detail` command, and is also accessible through the SNMP ifAlias object of IfXTable from IF-MIB (RFC 2233), and the XML API.

You can always configure a 255-character-long string regardless of the configured value of ifAlias size. Its value only affects the SNMP behavior.

Use the following commands to configure up to 255 characters associated with a port:

```
config port port_list description-string string
```

Use the following command to unconfigure the description-string setting:

```
unconfig port port_list description-string
```

Use the following command to control the accessible string size (default 64, per MIB) for the SNMP ifAlias object:

```
config snmp ifmib ifalias size [default | extended]
```

If you choose extended size option, the following warning will be displayed:

```
Warning: Changing the size to [extended] requires the use of increased 255 chars long
ifAlias object of ifXtable from IF-MIB(RFC 2233)
```

Port Groups

Port groups allow you to define ports by similar functionality or role. You can create port groups and use them to configure QoS profiles, ingress meters and flood rate-limiters for ports that share the same role. QoS commands that use port groups are updated automatically if the ports group is removed or if ports are added or removed from the group.

In addition, ExtremeCloud IQ Site Engine can create and manipulate port groups through the CoS MIB using SNMP. These groups are used to configure the ExtremeCloud IQ Site Engine-specified CoS resources that are mapped to QoS profiles, ingress meters, and flood rate-limiters. Since the ExtremeCloud IQ Site Engine CoS feature is port group-oriented, ExtremeXOS QoS entities (QoS profiles, ingress meter, and flood rate-limiting) that are configured on a per-port basis are not exposed through the CoS MIB using SNMP.

To create port groups, use the following command:

```
create ports group port_group
```

To delete port groups, use the following command:

```
delete ports group port_group
```

To add/delete ports from a port group, use the following command:

```
configure ports {group} port_group [[ add | delete ] port_list ]
```

To show port group information, use the following command:

```
show ports group {port_group}
```

Partitioning High Bandwidth Ports

5120 switches QSFP28 ports in Ethernet mode can be configured for 1x100G (default), 4x25G, or 4x10G.

5520, 5720, 7520-48Y-8C, 7520-48YE-8CE, 7520-48XT-6C, and 7720-32C switches QSFP28 ports in Ethernet mode can be configured for 1x40G (5520 default), 1x100G (except for 5520 switches), 2x50G, 4x25G or 4x10G. These ports are normally configured for stacking. To use these for Ethernet, you must disable stacking support.

Note
100G is not supported on 5520 switches QSFP28 ports.

7520 and 7720 switches' default QSFP28 port speed is 100G. The 7720-32C can also be configured for one 1G port with the 10506 QSA adapter.

Note
When replacing a Universal switch (for example, 5420 series or 5520 series) that is not in a stack, stacking-support should be disabled before applying a configuration where port partitioning is enabled on either of the Universal ports (U1 or U2).

Additionally, on 5520 (with 5520-VIM-4YE) switches, the 25G ports can be partitioned into 10G.

Note
This partition allows all ports to be converted either as 10G or as 25G and not both. To partition the 25G ports, apply the 4x10 option in the following command to the first VIM port (port 33 for 24-port models and port 57 for 48-port models).

The 5720 switch QSFP28 can be configured for one 100 Gb port (default), two 50 Gb ports, one

To partition the ports, enter the following command:

`configure ports [port_list | all] partition [1x100G | 1x40G | 2x50G | 4x10G | 4x25G]`

Additionally, on 5720 switches with 5720-VIM-6YE, the VIM ports can be partitioned to 3x1G, 3x10G or 3x25G. With 5720-VIM-2CE, the VIM ports can be partitioned to 4x10G, 4x25G, 1x40G or 1x100G.

Version 32.5 adds support for ports with different max speed capabilities to be part of the same Link Aggregation Group (LAG). This lets you change the speed and the auto-negotiation configuration of a port that is part of a LAG without unconfiguring and reconfiguring the LAG. This feature also supports dynamic repartitioning of the LAG ports without deleting and then recreating the LAG.

Note
Because of the nature of these ports at the physical layer level, the 10G side may show a remote or local linkup.

Flow Control

IEEE 802.3x Flow Control

With autonegotiation enabled, switches advertise the ability to support pause frames.

This includes receiving, reacting to (stopping transmission), and transmitting pause frames. However, the switch does not actually transmit pause frames unless it is configured to do so, as described below.

IEEE 802.3x flow control provides the ability to configure different modes in the default behaviors. Ports can be configured to transmit pause frames when congestion is detected, and the behavior of reacting to received pause frames can be disabled.

TX

You can configure ports to transmit link-layer pause frames upon detecting congestion. The goal of IEEE 802.3x is to backpressure the ultimate traffic source to eliminate or significantly reduce the amount of traffic loss through the network. This is also called lossless switching mode.

The following limitations apply to the TX flow control feature:

- Flow control is applied on an ingress port basis, which means that a single stream ingressing a port and destined to a congested port can stop the transmission of other data streams ingressing the same port that are destined to other ports.
- High volume packets destined to the CPU can cause flow control to trigger. This includes protocol packets such as, *EDP (Extreme Discovery Protocol)*, *EAPS (Extreme Automatic Protection Switching)*, *VRRP (Virtual Router Redundancy Protocol)*, and *OSPF (Open Shortest Path First)*.
- When flow control is applied to the fabric ports, there can be a performance limitation. For example, a single 1G port being congested could backpressure a high-speed fabric port and reduce its effective throughput significantly.

To configure a port to allow the transmission of IEEE 802.3x pause frames, use the following command:

```
enable flow-control tx-pause ports port_list|all
```

Note
To enable TX flow-control, RX flow-control must first be enabled. If you attempt to enable TX flow-control with RX flow-control disabled, an error message is displayed.

To configure a port to return to the default behavior of not transmitting pause frames, use the following command:

```
disable flow-control tx-pause ports
```

RX

You can configure the switch to disable the default behavior of responding to received pause frames. Disabling rx-pause processing avoids dropping packets in the switch and allows for better overall network performance in some scenarios where protocols such as TCP handle the retransmission of dropped packets by the remote partner.

To configure a port to disable the processing of IEEE 802.3x pause frames, use the following command:

```
disable flow-control rx-pause ports port-list | all
```

Note
To disable RX flow-control, TX flow-control must first be disabled. If you attempt to disable RX flow-control with TX flow-control enabled, an error message is displayed.

To configure a port to return to the default behavior of enabling the processing of pause frames, use the following command:

```
enable flow-control rx-pause ports port-list | all
```

Note
A port flap is expected when changing the pause settings. This occurs to propagate the new pause settings to the link partner, which requires restarting auto-negotiation. This process results in link flap events.

IEEE 802.1Qbb Priority Flow Control

Priority flow control (PFC) as defined in the IEEE 802.1Qbb standard is an extension of IEEE 802.3x flow control, which is discussed in IEEE 802.3x Flow Control on page 232.

When buffer congestion is detected, IEEE 802.3x flow control allows the communicating device to pause all traffic on the port, whereas IEEE 802.1Qbb allows the device to pause just a portion of the traffic while allowing other traffic on the same port to continue.

For PFC, when an ingress port detects congestion, it generates a MAC control packet to the connected partner with an indication of which traffic priority to pause and an associated time for the pause to remain in effect. The recipient of the PFC packet then stops transmission on the priority indicated in the control packet and starts a timer indicating when traffic can resume.

Traffic can resume in two ways:

- On the transmitting side, when the timer expires, traffic on that priority can resume.
- On the receiving side, once congestion is relieved, another PFC packet is generated to un-pause the priority so that traffic can resume.

Limitations

The following limitations are associated with this feature:

- PFC must be explicitly configured by the user.
- In order to support the signaling of congestion across the fabric, an enhanced fabric mode is required. This enhanced mode is not available on some older models of Summits (see the following supported platforms section). Also, this enhanced mode reduces the effective bandwidth on the fabric by a small amount (less than 5%).
- The fabric flow control packets take up some small amount of bandwidth on the fabric ports.

Supported Platforms

PFC is currently supported several port types, depending on the platform:

- 4120 switches: 25G, 100G uplink ports
- 4220 switches: 5G, 10G uplink ports
- 5120 switches: 10G uplink ports and U1/U2 in Ethernet mode
- 5320 switches (excluding 5320-16P-2MXT-2X): 10G uplink ports
- 5420F switches: 10G ports
- 5420M switches: 10G, 25G ports
- 5520 switches: 5G, 10G, 25G, 40G ports

- 5720 switches: 5G, 10G, 25G, 40G, 50G, 100G ports
- 7520 switches: 10G, 25G, 40G, 50G, 100G ports
- 7720 switches: 10G, 25G, 40G, 50G, 100G ports

If you attempt to enable PFC on unsupported ports, an error message is displayed. (See Abnormal Configuration Examples on page 237.)

Setting the Priorities

Priority is established for reception of PFC packets with a *QoS (Quality of Service)* profile value on the ExtremeXOS switch and for transmission with a priority value added to the PFC packet.

- QoS profile—Ingress traffic is associated with a QoS profile for assignment to one of eight hardware queues in the system that define how the traffic flows with respect to bandwidth, priority, and other parameters. By default, there are two QoS profiles (QP1 and QP8) defined in these supported platforms and PFC works with this default. To segregate the ingress traffic with more granularity, you will want to define other QoS profiles. The traffic that will be paused on reception of the PFC packet is associated with the hardware queue of the QoS profile that you specify.

  The QoS profile is also used to configure the fabric ports.
- Priority—When the hardware transmits a PFC packet, it uses the priority bits in the *VLAN (Virtual LAN)* header on the ingress packet to determine the priority to pause, if the ingress packet is tagged. You can specify this transmit priority independently from the QoS profile to associate it with the reception of a PFC packet thus giving flexibility in the configuration of the network. For untagged ingress packets, the hardware queue determines the priority in the transmitted PFC packet. (For additional information, see QoS Profiles on page 1469.

It is suggested that the priority in the VLAN header match the QoS profile priority when traffic ingresses at the edge of the network so that the traffic can be more easily controlled as it traverses through the network.

Fabric Port Configuration

This feature also configures the fabric between ingress and egress ports to optimize PFC behavior.

When the ingress and egress ports are located on separate nodes in a SummitStack, note that some older systems do not support the enhanced fabric mode required for PFC.

If any other ExtremeSwitching switch attempts to join the stack after the initial configuration of PFC, it is not allowed to join.

If your situation does not respond well to having flow control enabled on the fabric links, you can turn off flow control in the fabric by using the following command:

```
configure forwarding flow-control fabric [auto | off]
```

Configuring Priority Flow Control

With PFC, it is expected that both RX and TX be enabled or disabled.

- To enable PFC, use the following command:
  `enable flow-control [tx-pause {priority priority} | rx-pause {qosprofile qosprofile}] ports [all | port_list]`
- To disable PFC, use the following command:
  `disable flow-control [tx-pause {priority priority} | rx-pause {qosprofile qosprofile}] ports [all | port_list]`

Example

The network needs to transport FCoE (Fiber Channel over Ethernet) traffic which is intermixed with other more typical LAN traffic on the same Ethernet network. FCoE needs a lossless transport and PFC can be used to enable this. You define *QoS* profiles for all eight traffic priorities. At the network level, it is decided that FCoE traffic will be assigned to priority 3 (which corresponds to QP4) and the remaining traffic is assigned to one or more other priorities. For this example, it is also assumed that the priority bits in the incoming packets are also 3.

One mechanism that can be used for this classification is the use of Access Control Lists (ACLs) that match on the FCoE ethertypes (0x8906 and 0x8914) using the ethernet-type qualifier with an action of QoS profile QP4 for both rules. Other traffic can be classified to other priorities. Once this configuration is applied, FCoE is now separated from the other Ethernet traffic and is assigned a priority of 3 through the switch.

PFC is enabled at the ports that will see FCoE traffic, in this case, ports 1:1, 2:3, and 6:5.

Since FCoE is assigned to QP4, you would enable the receive PFC for QoS profile to be QP4 and, in this example, would also enable PFC with a transmit priority of 3. The enable commands would then read as follows:

```
enable flow-control tx-pause priority 3 ports 1:1,2:3,6:5
enable flow-control rx-pause qosprofile qp4 ports 1:1,2:3,6:5
```

The `show port flow-control rx-pause no-refresh` command displays the following information. With the no-refresh option the display shows the following information:

```
# show ports 1,5,9 flow-control rx-pauses no-refresh
Flow Control Frames Received
Port   Pause   PFC0   PFC1   PFC2   PFC3   PFC4   PFC5   PFC6   PFC7
        Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs
================================================================================
1          -      -      -      - 1234567 1234567    - 1234567     -
5          -      -      -      - 1234567 1234567    - 1234567     -
16:104 1234567    -      -      -       -       -    -       -     -
================================================================================
">" Name truncated, "-" rx-pause not enabled, "." Counter not available
```

Without the no-refresh option the display refreshes until interrupted by the console operator pressing `Escape`:

```
# show ports 1,5,9 flow-control rx-pauses
Flow Control Frame Monitor                    Sat Aug 18 19:35:12 2012
```

```
Port   Pause   PFC0   PFC1   PFC2   PFC3   PFC4   PFC5   PFC6   PFC7
        Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs   Rcvs
================================================================================
1          -      -      -      - 1234567 1234567    - 1234567     -
5          -      -      -      - 1234567 1234567    - 1234567     -
16:104 1234567    -      -      -       -       -    -       -     -
================================================================================
">" Name truncated, "-" rx-pause not enabled, "." Counter not available
Spacebar->Toggle screen 0->Clear counters U->Pageup D->Pagedown ESC->exit
```

The new `show port flow-control tx-pause no-refresh` CLI command displays the following information. With the no-refresh option the display shows the following information:

```
# show ports 1,5,9 flow-control tx-pauses no-refresh
Flow Control Frames Transmitted
Port   Pause   PFC0   PFC1   PFC2   PFC3   PFC4   PFC5   PFC6   PFC7
       Xmts    Xmts   Xmts   Xmts   Xmts   Xmts   Xmts   Xmts   Xmts
================================================================================
1         -       -      -      - 1234567 1234567    - 1234567    -
5         -       -      -      - 1234567 1234567    - 1234567    -
16:104 1234567    -      -      -       -       -    -       -    -
================================================================================
">" Name truncated, "-" tx-pause not enabled, "." Counter not available
```

Without the no-refresh option the display refreshes until interrupted by the console operator pressing the escape key:

```
# show ports 1,5,9 flow-control tx-pauses
Flow Control Frame Monitor Sat Aug 18 19:35:12 2012
Port   Pause   PFC0   PFC1   PFC2   PFC3   PFC4   PFC5   PFC6   PFC7
        Xmts   Xmts   Xmts   Xmts   Xmts   Xmts   Xmts   Xmts   Xmts
================================================================================
1          -      -      -      -  1234567 1234567   -  1234567    -
5          -      -      -      -  1234567 1234567   -  1234567    -
16:104 1234567    -      -      -       -       -   -       -    -
================================================================================
">" Name truncated, "-" tx-pause not enabled, "." Counter not available
Spacebar->Toggle screen 0->Clear counters U->Pageup D->Pagedown ESC->exit
```

Once this configuration is complete, if a switch ingress port detects congestion, it will send PFC packets to the remote link partner and will respond to PFC packets from the remote link partner by stopping transmit.

Abnormal Configuration Examples

Examples of abnormal configuration scenarios.

- If you attempt to configure PFC on a port that does not support it, an error message similar to the following is issued and you will be informed that PFC cannot be configured on that port:

  ```
  enable flow-control tx-pause priority 3 port 1:1
  Error: Port 1:1 does not support Priority Flow Control.
  ```

- When PFC is enabled on a port, IEEE 802.3x will be disabled. If, after enabling PFC, you try to modify RX or TX pause parameters, an error message similar to the following will be issued explaining the dependency on PFC configuration:

  ```
  BD8810.1# enable flow-control tx-pause port 1:1
  Error: Priority Flow Control is currently enabled on port 1:1 and is mutually
  ```

```
exclusive with TX and RX pause configuration. TX and RX pause configuration cannot be
done until PFC is disabled on this port.
```

Turning off Autonegotiation on a Gigabit Ethernet Port

In certain interoperability situations, you need to turn autonegotiation off on a fiber gigabit Ethernet port. Although a gigabit Ethernet port runs only at full duplex, you must specify the duplex setting. The 10 Gbps ports do not autonegotiate; they always run at full duplex and 10 Gbps speed.

Running Link Fault Signal

The 10 Gbps ports support the Link Fault Signal (LFS) function. This function, which is always enabled, monitors the 10 Gbps ports and indicates either a remote fault or a local fault. The system then stops transmitting or receiving traffic from that link. After the fault has been alleviated, the system puts the link back up and the traffic automatically resumes.

The Extreme Networks implementation of LFS conforms to the IEEE standard 802.3ae-2002.

Note
To display the part number of the switch in a SummitStack, use the `show slot` *`slot_number`* command.

Although the physical link remains up, all Layer 2 and above traffic stops.

The system sends LinkDown and LinkUp traps when these events occur. Additionally, the system writes one or more information messages to the syslog, as shown in the following example:

```
09/09/2004 14:59:08.03 <Info:vlan.dbg.info> MSM-A: Port 4:3 link up at
10 Gbps speed and full-duplex
09/09/2004 14:59:08.02 <Info:hal.sys.info> MSM-A: 4:3 - remote fault recovered.
09/09/2004 14:59:05.56 <Info:vlan.dbg.info> MSM-A: Port 4:3 link down
due to remote fault
09/09/2004 14:59:05.56 <Info:hal.sys.info> MSM-A: 4:3 - remote fault.
09/09/2004 15:14:12.22 <Info:hal.sys.info> MSM-A: 4:3 - local fault
recovered.
09/09/2004 15:14:11.35 <Info:vlan.dbg.info> MSM-A: Port 4:3 link up at
10 Gbps speed and full-duplex
09/09/2004 15:13:33.56 <Info:vlan.dbg.info> MSM-A: Port 4:3 link down
due to local fault
09/09/2004 15:13:33.56 <Info:hal.sys.info> MSM-A: 4:3 - local fault.
09/09/2004 15:13:33.49 <Info:vlan.dbg.info> MSM-A: Port 4:3 link down
due to local fault
```

On ExtremeSwitching switches, after disabling the 10 Gbps ports, the following message appears in the Syslog:

```
08/26/2008 06:05:29.29 Port 1 link down - Local fault
```

Note
A link down or up event may trigger Spanning Tree Protocol topology changes or transitions.

Turn off Autopolarity

The autopolarity feature allows the system to detect and respond to the Ethernet cable type (straight-through or crossover cable) used to make the connection to the switch port or an endstation.

Turning off autopolarity applies only to the 10/100 BASE-T ports on the switch and copper medium on ExtremeSwitching combination ports.

When the autopolarity feature is enabled, the system causes the Ethernet link to come up regardless of the cable type connected to the port. When the autopolarity feature is disabled, you need either a crossover or a straight-through cable depending on the native polarity of the switch (it may vary across platforms) and the type of equipment being connected to the port. When disabling autopolarity, you have the option to set the port's polarity or Medium-Dependent Interface (MDI) mode to straight-through (MDI) or crossover (MDIX) mode irrespective of the native MDI mode of the port. The autopolarity feature is enabled by default.

Under certain conditions, you can choose to turn autopolarity off and set a specific MDI mode on one or more ports.

- To disable or enable autopolarity detection, use the following command:

  `configure ports` *`port_list`* **`auto-polarity`** `[`**`on`** `|` **`off`** `{` **`mdi-mode`** `[` **`default`** `|` **`mdi`** `|` **`mdix`**`]}]`

  Where:

  - port_list—Specifies one or more ports on the switch
  - off—Disables the autopolarity detection feature on the specified ports
  - on—Enables the autopolarity detection feature on the specified ports
  - mdi-mode—Medium-Dependent Interface (MDI) mode
  - default—Use the platform's default MDI mode
  - mdi—Force MDI mode to be straight-through
  - mdix—Force MDI mode to be crossover

  The following example turns autopolarity off and sets the polarity as crossover for ports 5 to 7 on an ExtremeSwitching switch:

  ```
  configure ports 5-7 auto-polarity off mdi-mode mdix
  ```

- When autopolarity is disabled on one or more Ethernet ports, you can verify that status using the command:

  ```
  show port information detail
  ```

Flow Monitor

Flow Monitor provides IP Flow Information Export (IPFIX) and K-Mirror support based on RFC 7011 to collect and export flow records for Universal platforms. Flow Monitor builds on IPFIX by providing additional metering and exporting processes. Flow Monitor enables switches to gather information about flows through different ports on each switch and export the flow records (via UDP) to a remote collector(s). This

information can be used to analyze application and network usage, and monitor real-time network infrastructure.

Flow Monitor supports up to eight external collectors.

Supported Features

The following table lists feature and version support for Flow Monitor:

Table 22: Flow Monitor Feature Support

Feature	Switch Engine Version
FTFP match rules, Keygen, Group/Profile, Collector, and Template structures	32.2 and later
Packet sampling (aka, K-mirror) via local port or ERSPANv3	32.2 and later
Flow state transition (TCP SYN/ACK/FIN)	32.2 and later
Hardware-based flow export in RFC7011 IPFIX format	32.2 and later
Hardware aging and flushing of the flow table	32.2 and later
Status to ingress field processor allows flow status to be an input to ACL rule matching	32.2 and later

Supported Platforms

This feature is supported on 5420, 5520, and 5720 series switches.

Limitations

- Flow Monitor is not compatible with legacy IPFIX.
- Legacy ExtremeXOS IPFIX templates for non-IP packets are not supported.
- Transport of Flow Monitor records to Collector(s) is UDP only.
- Collectors must be reachable on an IPv4 subnetwork and are not reachable through the IPv6 network.
- Histograms are not supported in version 32.2.
- Support for an internal collector is not provided.
- Datagram Transport Layer Security (DTLS) is not supported.

Enabling Flow Monitor

Flow Monitor is disabled by default. When enabled, flow collection and exporting occurs on configured keys in enabled groups. The `enable flowmon` command enables the Flow Monitor feature.

To enable a Flow Monitor group, use the following command: `enable flowmon` **`group`** *`group_name`*.

Note
While a group is being enabled, it cannot be modified except to add or delete keys.

Note
Before a group can be enabled, it must have at least one key and one collector added to it. A template key portion is present when a key has been added.

To disable Flow Monitor globally, use the following command: `disable flowmon`.

Note
Modification rules for groups, keys, and collectors remain active after Flow Monitor is disabled. A new group can be created and configured with its parameters, collector, template portions (defines the elements to be exported), and added keys. The group can also be enabled or disabled.

Note
Enabling a group while Flow Monitor is disabled will program the hardware, but the flow collection for the group will be disabled.

To disable a Flow Monitor group, use the following command: `disable flowmon` **`group`** *`group_name`*.

Configuring Flow Monitor Collectors

A Flow Monitor collector must be created before it can be configured. To create a Flow Monitor collector, use the `create flowmon` **`collector`** *`collector_name`* command. The *`collector_name`* can contain up to 32 characters.

Up to eight external collectors are supported. The created collector must be configured before it can be added to a group, and a collector can be used by many groups.

Note
The system will reject any attempt to create a collector that already exists.

To export flow records using the Flow Monitor, you must first configure a collector. To configure a collector, use the `configure flowmon` **`collector`** *`collector_name`* command.

Most of a collector's configuration cannot be changed or deleted while the collector is added to any group. You can change the refresh time at any time.

If any of the specified parameters cannot be set, or if there is an error honoring the configuration request, the configuration command will be ignored. While this scenario will affect configured parameter values, the state of the UDP socket might not be affected.

To delete a Flow Monitor collector, use the `delete flowmon` **`collector`** *`collector_name`* command.

Note
The system will reject any attempt to delete a collector that does not exist.

Configuring Flow Monitor Keys

A Flow Monitor key must be created before it can be configured. To create a Flow Monitor key, use the `create flowmon` **`key`** *`key_name`* command. The *`key_name`* can contain up to 32 characters.

Flow keys are used to identify the flows to be tracked by a specified flow group. Flows keys are a combination of parameters that can be generic, for example, 'ipv4', or more specific, for example, 'ipv4 src-port 80 dst-port 122'.

Note
The system will reject any attempt to create a key that already exists.

To configure a Flow Monitor key with IPv4 parameters, use the `configure flowmon` **`key`** *`key_name`* **`ipv4`** command.

To configure a Flow Monitor key with IPv6 parameters, use the `configure flowmon` **`key`** *`key_name`* **`ipv6`** command.

Note
If a specific port list indicates a slot where Flow Monitor is not supported, the command will be rejected.

To delete a Flow Monitor key, use the `delete flowmon` **`key`** *`key_name`* command.

If a key is added to the group, it cannot be deleted using the **`delete flowmon key`** command until the key is deleted from the group. The related template key is disassociated and only deleted when there are no more keys or groups referencing it.

Note
The system will reject any attempt to delete a key that does not already exist.

Configuring Flow Monitor Groups

A Flow Monitor group must be created before it can be configured. To create a Flow Monitor group, use the `create flowmon` **`group`** *`group_name`* command. The *`group_name`* can contain up to 32 characters.

Note
The system will reject any attempt to create a group that already exists.

To configure a created group where Flow Monitor sends information, use the `configure flowmon` **`group`** *`group_name`* command.

To configure a relationship between a Flow Monitor collector and a group, use the `configure flowmon` **`group`** *`group_name`* [**`add`** | **`delete`**] **`collector`** *`collector_name`* command.

Only one collector can be added to a group, but the same collector can be added to multiple groups.

Note
The system will reject any attempt to add a collector to a group that already contains a collector. The system will also reject any attempt to delete a collector from a group that was not previously added to the group.

To configure a relationship between a Flow Monitor key and a group, use the `configure flowmon` **`group`** *`group_name`* [**`add`** | **`delete`**] **`key`** *`key_name`* command. While a key is being added to a group, the key cannot be modified.

Flow Monitor creates a template key portion that matches the key. If a different template key portion is related to the group, then the add fails. Otherwise, the template key portion is related to the group.

If the group is already enabled, the key is installed in the hardware. Keys can be added to or deleted from a group regardless of the state of the group.

When a key is to be deleted from a group, the key is disassociated from the key template portion. The key is then disassociated from the group. If the group has no more keys associated with it and the group is disabled, the group is then disassociated from the template key portion.

To delete a Flow Monitor group, use the `delete flowmon` **`group`** *`group_name`* command.

An enabled group cannot be deleted and the system will reject any attempt to do so. Groups must be disabled before deleting.

Note
The system will reject any attempt to delete a group that does not already exist.

Clearing Flow Monitor Counters

To clear all Flow Monitor groups, use the following command:

```
clear counters flowmon
```

Displaying Flow Monitor Information

You can view Flow Monitor information on ports with the following commands:

The `show flowmon` command shows whether Flow Monitor is in enabled or disabled state, the number of groups, number of keys, and number of collectors. It also shows the current total number of flows learned in each hardware domain.

The `show flowmon` **`collector`** [*`collector_name`* | **`all`**] **`detail`** command shows Flow Monitor collector information.

When detail is entered, the source and destination IPv4 addresses, source and destination UDP ports, VR name, and export MTU size are displayed.

Note
If *collector_name* is not entered, then a list of collectors is displayed.

The `show flowmon` **`group`** *`group_name`* `|` **`all`**`]` **`detail`** command shows Flow Monitor group information. For each Flow Monitor group, the group name and parameters are shown.

When detail is entered, the assigned collector name, complete template name, template key portion name, and a list of key names, dependent keys and containing groups are shown. In addition, the group state and whether the group is enabled or disabled are shown. The group state is defined as follows:

- Configuring – The user is configuring the group, and the group is not enabled.
- GroupInstall – The group is being installed.
- KMirrorInstall – The K-mirror rule is being installed for the group.
- KeysInstall – The rules related to the keys that are added to the group are being installed.
- Ready – The group is fully installed and awaiting hardware activation.
- CollectorActivate – The collector is being informed of the complete templates associated to the group.
- GroupActivate – The group is being activated.
- Active – The group is fully active, and flows are being collected and advertised to the collector.

When all is entered, then all groups are shown with full detail.

The `show flowmon` **`group`** *`group_name`* **`statistics no-refresh wide`** command shows Flow Monitor group statistics information.

When *group_name* is entered, only the statistics for that group are shown. When not entered, all groups are shown using the **Page Up** and **Page Down** options.

Each group name is displayed along with the number of active flows, the number of flows that have aged out, the number of flows that were not learned because the group flow limit was exceeded, the number of flows learned on the group, and the number of flows that were not learned because the flow table was full.

When no-refresh is provided, the specified group or groups are displayed one time with their current values.

If no-refresh is not entered, the display is updated with refreshed values every five seconds. This display operates as a standard updatable Switch Engine statistics display, providing **Page Up**, **Page Down**, **Exit**, and **Clear** options.

When no-refresh is entered, the specified group or groups are displayed one time with their current values.

When wide is entered, a wider version of the display is shown. If wide is not entered, the counters are limited to a display width that fits into 80 display columns. When wide is provided, the full 64-bit counter value can be displayed (about 21 columns per integer value).

The `show flowmon` **`group`** *`group_name`* **`template detail`** command displays Flow Monitor group template information. The complete template name is shown with the template key portion name.

When detail is entered, the Element IDs are shown in text form.

The `show flowmon` **`key`** *`key_name`* command shows Flow Monitor key information. For each key, the name, type, associated group name (if applicable), and the 'before' or after key name (if applicable) are displayed.

When all is entered and detail is not specified, all keys are shown in tabular format.

When detail is entered, the IPv4 or IPv6 address and mask, source or destination UDP or TCP port, protocol or next header, and specified list of ports to which the key is applied are shown. If none are specified, all values are shown.

Configuring Switching Mode—Cut-through Switching

Default Switching Mode

All platforms use store-and-forward by default. The platforms are also capable of supporting cut-through forwarding to reduce latency. Store-and-forward switching requires the complete receipt of a packet prior to transmitting it out the interface. The packet is stored in its entirety in packet memory and can be validated via the frame CRC by the switch prior to forwarding it to the next hop.

Cut-through switching allows the switch to begin transmitting a packet before its entire contents have been received thereby reducing the overall forwarding latency for large packet sizes.

Cut-through forwarding mode is supported on 10G, 25G, 40G, 50G, and 100G ports on Extreme Networks 7520 and 7720 series switches.

The following limitations apply to the cut-through switching feature:

- Error packets may be forwarded when using cut-through mode. These packets need to be detected and discarded by one of the downstream switches, routers, or the ultimate end station.

In some circumstances, store-and-forward is automatically used. Following are examples:

- Cut-through mode cannot be achieved when switching a packet internally from a low-speed front-panel port (1G or 10G) to a higher-speed fabric port. In this case, store-and-forward switching will automatically be used. However, cut-through switching can be used when switching between equal speed ports or from 100-G or less to 40G ports or from 50G or less to 25G or 10G ports.

- Store-and-forward is used for packets that are switched to multiple egress ports in scenarios such as *VLAN* flooding and multicast.
- Store-and-forward is used whenever the egress interface is congested including when *QoS* rate shaping is in effect.

Configuring Switching Mode

You can change or view the default switching mode.

- To configure the switching mode, use the following command:
  `configure forwarding switching-mode [cut-through | store-and-forward]`
- To display the switching mode settings, use the following command:
  `show forwarding configuration`

Using the Precision Time Protocol

IEEE1588v2 (also known as Precision Time Protocol, or PTP) is an industry-standard protocol that enables the precise transfer of frequency and time to synchronize clocks over packet-based Ethernet networks.

The locally available clock on each network device synchronizes with a grandmaster clock by exchanging timestamps that contain sub-nanosecond granularity. This allows them to deliver very high accuracy to ensure the stability of base station frequency and handovers. The timestamps between master and slave devices are exchanged through PTP event packets. The ExtremeXOS 1588v2 implementation uses the IPv4/UDP transport mechanism PTP packets.

The Precision Time Protocol is currently available only on 7520-48Y and 7720-32C series switches. For this router, accurate synchronization of base stations to nanoseconds accuracy is critical to minimize service disruptions and eliminate dropped connections as calls move between adjacent cells.

Overview of PTP

The IEEE 1588v2 Precision Time Protocol (PTP) defines a packet-based time synchronization method that provides frequency, phase, and time-of-day information with nanoseconds level of accuracy.

PTP relies on the use of carefully time-stamped packets to synchronize one or more slave clocks to a master clock. Synchronous time information is distributed hierarchically, with a grandmaster clock at the root of the hierarchy.

The grandmaster provides the time reference for one or more slave devices. These slave devices can, in turn, act as master devices for further hierarchical layers of slave devices.

To determine the master-slave hierarchy, a Best Master Clock (BMC) algorithm is used. This algorithm determines which clock is the highest quality clock within a network. The clock elected by BMC (the master clock) then synchronizes all other clocks (slave clocks) in the network. If the BMC is removed from the network or is determined by the BMC algorithm to no longer be the highest quality clock, the algorithm then redefines the new BMC and adjusts all other clocks accordingly. No administrator input is needed for this readjustment because the algorithm provides a fault tolerant behavior.

Synchronizing time across a network requires two essential functions: the measurement of delays and the distribution of time information. Each node is responsible for independently determining the delays across the network links from it to its link partners. Once this is accomplished, periodic time synchronization messages may be sent from the grandmaster clock device to the slave clock devices. Link-based delays wander over time, so periodic delay measurements are required. Because these delays vary slowly, the period between link delay measurements is typically in the order of seconds.

A PTP network must have a grandmaster clock reference and a slave. Between a master and a slave, a PTP network may have multiple boundary clocks, transparent clocks, and non-PTP bridges.

Figure 7 illustrates a typical PTP network hierarchy.

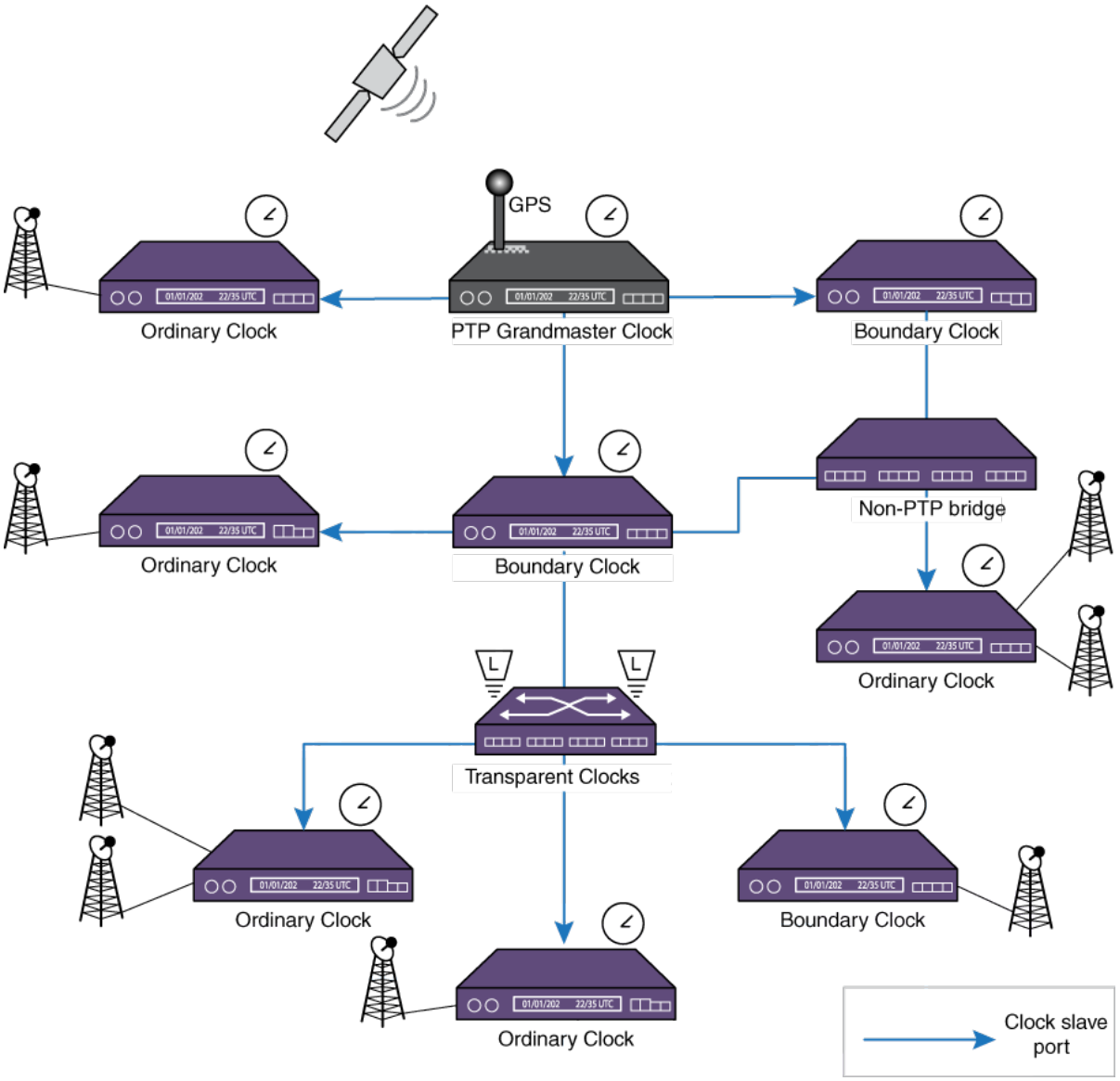


Figure 7: PTP Network Hierarchy

Ordinary clocks are devices with only one PTP port. The grandmaster clock is an ordinary clock acting as a master.

Note
A PTP port is a logical interface (*VLAN* / IP interface).

Boundary clocks are switches with one or more PTP ports. One PTP port of a boundary clock can act as a slave to a master clock in the network, and the rest of the PTP ports can act as a master for the downstream nodes.

Transparent clocks correct the delays for PTP messages in the correction field.

End-to-end transparent clocks accumulate the residence time in the CorrectionField of the PTP messages. End-to-end transparent clocks do not participate directly in time synchronization with the master clock. The CorrectionField of Sync, Delay Request, and Delay Response messages are updated by the end-to-end transparent clocks at each hop. The Signaling and Management messages are not updated by transparent clocks. In a typical setting, boundary and slave clocks are separated by one or more end-to-end transparent clocks that accumulates the residence time in the CorrectionField.

The residence time is defined as the delay between the reception and the transmission of packets through the device. The accumulated CorrectionField value is used by boundary or slave clocks for delay compensation in the time offset correction.

Basic Synchronization

The following event flow describes basic synchronization of PTP:

1. The master sends a Sync message to the slave, notes the time (t1) it was sent, and embeds the t1 time in the message.
2. The slave receives the Sync message, and notes the time it is received (t2).
3. The master conveys to the slave the timestamp t1 either by (1) embedding the timestamp t1 in the Sync message (this requires hardware processing for highest accuracy and precision, and is called as one step clocking) or by (2) embedding the timestamp t1 in a Follow_Up message (this is called as two step clocking).
4. The slave sends a DelayReq message to the master, and notes the time (t3) it was sent.
5. The master receives the DelayReq message, and notes the time it is received (t4).

6. The master embeds the t4 timestamp in a DelayResp message to the slave.

   At the conclusion of this exchange of messages, the slave possesses all four timestamps. You can use these timestamps to compute the offset of the slave's clock with respect to the master, and the mean propagation time of messages between the two clocks.

   The formula to compute the offset is as follows:

   - master_to_slave_delay = offset + delay = t2 – t1
   - slave_to_master_delay = offset − delay = t4 – t3
   - one_way_delay = (master_to_slave_delay + slave_to_master_delay) / 2

   The calculation for the offset from the master is as follows: offset_from_master = t2 - t1 - one_way_delay.

   Note
   The above explanation for offset calculation is provided simply for understanding the basic PTP protocol. The actual calculations involve many more variables.

The computation of offset and propagation time assumes that the master-to-slave and slave-to-master propagation times are equal. Any asymmetry in propagation time introduces an error in the computed value of the clock offset.

End-to-End Transparent Clocks Between Master And Slave

PTP defines the notion of End-to-end transparent clocks which do not participate in time synchronization with master clock.

Rather, they simply accumulate the residence time of PTP event messages such as Sync/DelayReq that transit the switch. The residence time is updated in the CorrectionField of these messages.

The transit delay in the link between the hops are not accounted in the CorrectionField by the End-to-end transparent clocks.

PTP Slave Clock Adjustments

The following sections identify factors and components involved in Slave Clock adjustment. The actual values for the clock adjustment are determined from a Servo Algorithm described in .

PTP Slave Time Correction

Each PTP network element maintains a PTP-free running time-of-day counter used as a basis for generating recovered clock signals and computing latencies, offsets and drift rates. The free-running counter runs on the local system clock, asynchronously to the clocks maintained by the other members of the PTP network. Correction factors are applied to the free-running clock in order to arrive at a local time value that is synchronized to the grand master clock. The PTP free-running time value consists of a 32-bit nanoseconds field, and a 48-bit seconds field. Every second the nanoseconds [31:0] field is rolled over and the 48-bit seconds [47:0] counter is incremented. The

PTP network element uses the information calculated from the master clock's sync messages to perform local clock adjustments.

Drift Adjustment

If the trend of slave offset values calculated from the Sync Messages continues to increase or decrease over time, the local reference clock that increments the free-running counter is operating at a rate slightly slower or faster than the master reference. You can make a drift adjustment to the free-running counter by slightly increasing or decreasing the rate at which the counter increments. Doing so locks the frequency of the counter to the master reference (syntonization). Syntonization is the adjustment of a clock signal to match the frequency, but not necessarily the phase, of another clock signal.

The drift adjustment calculation is done as follows:

Let, X = syncEventIngressTimestamp
Y = correctedMasterEventTimestamp

```
                    Xn – X0

drift Adjustment = ---------
                    Yn – Y0
```

Where, n = number of sync interval separating time stamps (n>0).

Syntonization may also be achieved using SyncE.

Offset Adjustment

Once the drift rate has been measured and compensated for correctly, the slave clock offset should remain fairly constant at each Sync interval. Ideally, once an offset is computed and put in place, it is only rarely changed. The offset is applied to the local time value to synchronize the local time with the master's.

Ultimately, the slave clock uses the drift and offset adjusted counter to generate a usable clock signal externally. Through PTP, each slave free-running counter is both frequency and time-of-day locked to the master clock. The slave clock uses the free-running clock and phase information to generate a frequency and phase aligned clock signal traceable to the master clock.

PTP Clock Servo Algorithm

PTP as a protocol basically gives the timestamp values based on its messages and operation for calculating offset and drift adjustment. The problem is that the Slave clock cannot be simply corrected by setting the free running counter to a new value. If done this way, there would be time intermission or time back scenarios, and inaccurate synchronization. Thus the change to the slave clock should be brought in gradually while taking several other factors into account that affect the working of PTP:

- Packet Error (if there is an error in time stamp)
- Extended Packet Loss (an outage scenario)
- Packet Delay Variation (network load increase)

Timestamps provided by PTP are used as an input to the Servo Algorithm. This algorithm addresses all of above mentioned network impairments (PDV being the most critical one) and gives an output for the amount of adjustment that should be made to the local slave clock. 1588v2 Specification does not define the Servo part. The Servo Algorithm's goal is to achieve zero time difference between the master and the slave, and that the frequency of the slave clock and master clock are locked (meaning the ratio should be constant).

Hybrid Networks

The Precise Time Protocol (PTP) synchronizes the network by transferring the master clock information in the form of timestamps in the PTP messages (Sync/FollowUp/DelayReq/DelayResp). In the slave clock, the clock offset is computed through the reception of PTP messages that carry master clock as timestamps.

In practice, a network could employ multiple synchronization methods in the same network. SyncE transfers the frequency of the reference clock through the ethernet's physical layer. The frequency recovered from the Synchronous Ethernet is highly accurate when compared to the frequency recovered through PTP messages. However, SyncE does not carry the Time-of-Day (TOD), or the Phase information of the clock as PTP does. Networks that employ SyncE and PTP for synchronization can leverage the accuracy of time transfer through PTP by using SyncE. Such Hybrid networks use SyncE for frequency transfer, and PTP for Phase/Time-of-Day transfer.

PTP Time-source

Each node in the Hybrid network recovers and transfers primary reference frequency using SyncE and phase/ToD using PTP. The PTP implementation usually includes a local reference clock at each boundary and ordinary clocks. The PTP protocol synchronizes this local reference clock to the Grandmaster clock by correcting its offset. In Hybrid networks, since frequency of the Primary reference is already available through clock recovered from SyncE, this recovered clock is used as a local reference clock. The timestamps received through the PTP event messages are processed and the Phase/ToD information is recovered.

The PTP protocol does not specify the source of time (frequency and/or Phase/ToD), or the method of clock recovery. This is left to the implementation. For boundary and ordinary slave clocks, the source of time is through PTP messages. However, since PTP messages carry timestamps that can be used to recover both frequency and phase/ToD, the implementations separate this aspect of recovery, and perform only phase/ToD recovery on the received PTP messages. The frequency recovery from the received PTP messages is not performed. The PTP implementation uses the recovered clock from SyncE as a frequency time-source.

Supported PTP Features

The following PTP features are supported in this release:

- Boundary Clock and Ordinary Clock
- End-to-End Transparent Clock
- IPv4 Unicast-UDP transport

- PTP protocol 1-step mode with end-to-end delay mechanism.
- Unicast masters.

7520 and 7720 Supported PTPv2 Profiles

PTPv2 supports the following profiles on 7520 and 7720 switches.

G.8265.1 Profile		
Parameter	**Value**	**Description**
Domain number	4	PTP domain identifier
Announce interval	1	Announce message interval
Announce timeout	3	3 × announce interval
Sync interval	0	Sync message interval
Delay interval	0	Delay request interval

G.8275.1 Profile		
Parameter	**Value**	**Description**
Domain Number	24	PTP domain identifier
Announce Interval	1	Announce message interval
Announce Timeout	3	3 × announce interval
Sync Interval	-3	Sync message interval
Delay Interval	0	Delay request interval

G.8275.2 Profile		
Parameter	**Value**	**Description**
Domain Number	44	PTP domain identifier
Announce Interval	1	Announce message interval
Announce Timeout	3	3 × announce interval
Sync Interval	0	Sync message interval
Delay Interval	0	Delay request interval

SMPTE ST 2059-2:2021 Profile		
Parameter	**Value**	**Description**
Domain Number	127	PTP domain identifier
Announce Interval	1	Announce message interval
Announce Timeout	3	3 × announce interval

SMPTE ST 2059-2:2021 Profile		
Parameter	Value	Description
Sync Interval	-3	Sync message interval
Delay Interval	0	Delay request interval

AES67-2018 Profile		
Parameter	Value	Description
Domain Number	0	PTP domain identifier
Announce Interval	0	Announce message interval
Announce Timeout	3	3 × announce interval
Sync Interval	-3	Sync message interval
Delay Interval	0	Delay request interval

The following example commands configure various PTPv2 profiles:

```
create network-clock ptp boundary domain 44

configure network-clock ptp boundary priority1 128

configure network-clock ptp boundary priority2 128

configure network-clock ptp boundary add vlan v100 one-step

configure network-clock ptp boundary announce interval 1 vlan v100

configure network-clock ptp boundary announce timeout 3 vlan v100

configure network-clock ptp boundary delay-request-interval -6 vlan v100

configure network-clock ptp boundary sync-interval -6 vlan v100

configure network-clock ptp boundary add vlan v200 one-step

configure network-clock ptp boundary announce interval 1 vlan v200

configure network-clock ptp boundary announce timeout 3 vlan v200

configure network-clock ptp boundary delay-request-interval -6 vlan v200

configure network-clock ptp boundary sync-interval -6 vlan v200

enable network-clock ptp boundary
```

Limitations of PTP

The following are the limitations of the implementation of PTP:

- Layer 2 transport is not supported.
- IPv6-UDP transport is not supported.

- Multicast event messages are not supported.
- Peer-to-peer delay mechanism is not supported.
- PTP datasets are not maintained for end-to-end transparent clocks.
- Domain number cannot be assigned to end-to-end transparent clocks.
- Boundary clock does not support synchronizing clocks across multiple domains.
- Distributing clock frequency recovered from SyncE or from BITS or from a TDM port over PTP is not supported.
- Ordinary clock master (Grandmaster) mode is not supported.
- Synchronizing system time with the time recovered from PTP event messages is not supported.
- Time of Day (ToD) output and inputs are not supported.
- Unicast message negotiation on clock ports is not supported.
- PTP cannot be used if network clock source is configured as BITS.

The following limitation also apply to 7520 and 7720 series switches:

- Stacking with PTP is not supported.

Configuring and Displaying PTP Clocks and Data Sets

PTP Transparent clock

A PTP Transparent clock updates the residence time of the PTP event packets that transit the switch. The switch supports End-to-End delay mechanism and accounts for the residence time for Sync and DelayReq packets in the switch.

- To create or delete an End-to-End Transparent clock, use the following command:

```
create network-clock ptp end-to-end-transparent
delete network-clock ptp end-to-end-transparent
```

- To add PTP capable front-panel ports for End-to-End Transparent clock, use the following command:

```
configure network-clock ptp end-to-end-transparent [add | delete] ports port_list {one-step}
```

- To display the End-to-End Transparent clock configuration, use the following command:

```
show network-clock ptp end-to-end-transparent ports port_list {detail}
```

- To enable/disable the End-to-End Transparent clock configuration on the front-panel ports, use the following commands:

```
enable network-clock ptp end-to-end-transparent ports port_list
disable network-clock ptp end-to-end-transparent ports port_list
```

PTP Boundary/Ordinary Clocks

A PTP Boundary or Ordinary clock synchronizes to the master clock through the reception of the PTP event packets. To reconfigure clocks to a different domain, the

existing configuration must be deleted. The Boundary and Ordinary clocks can be configured to operate on a single PTP domain.

- To create or delete the Boundary or Ordinary clock, use the following commands:

  ```
  create network-clock ptp [boundary | ordinary] [{domain domain_number} | {profile profile_name}] [g8265.1 | g8275.1 | g8275.2 | aes67-2018 | st-2059-2] | end-to-end-transparent]
  ```

  ```
  delete network-clock ptp [boundary | ordinary]
  ```

- Enable or disable the Boundary or Ordinary clock, use the following commands:

  ```
  enable network-clock ptp [boundary | ordinary]
  ```

  ```
  disable network-clock ptp [boundary | ordinary]
  ```

  Note
  After you enable a boundary clock, you cannot create an ordinary clock. However, you can delete the boundary clock instance and create a new one in order to change the domain number.

  To create an ordinary clock instance in the switch that has the boundary clock instance enabled, delete the boundary clock instance, save the configuration and reboot the switch. After the reboot, you can create and enable the ordinary clock instance. Similarly, to create and enable a boundary clock in a switch that has an ordinary clock enabled, delete the ordinary clock instance, save the configuration and reboot the switch. After the reboot you can create and enable a boundary clock. The following message is displayed when you create the boundary clock instance in a device with no prior clock instances:

  ```
  Warning: The ordinary clock cannot be created after enabling the boundary clock. A delete followed by save and reboot are required to create the ordinary clock.
  ```

  After you enable a boundary clock instance, if you delete the instance and try to create an ordinary clock instance, the above message is displayed as an error, and the ordinary clock instance is not created.

- To configure priority1 and priority2 values of the Boundary and Ordinary clock, use the following commands:

  ```
  configure network-clock ptp [boundary | ordinary] priority1 priority
  ```

  ```
  configure network-clock ptp [boundary | ordinary] priority2 priority
  ```

- To display the datasets such as Port, Time-properties and Parent of the Ordinary or Boundary clock, use the following commands:

  ```
  show network-clock ptp ordinary {parent | port | time-property}
  ```

  ```
  show network-clock ptp boundary {parent | port | time-property}
  ```

PTP Boundary/Ordinary Clock Ports

The Boundary and Ordinary clocks operate in 1-step protocol mode. An Ordinary clock can have at most one clock port in slave mode. A Boundary clock can have multiple clock ports in master or slave modes. Multiple unicast master or slave entries can be added to the clock ports.

- Add or remove a slave clock port to an Ordinary clock.

  ```
  configure network-clock ptp ordinary add {vlan} vlan_name {one-step} slave-only
  ```

```
configure network-clock ptp ordinary delete {vlan} {vlan_name}
```

- Add or remove a clock port to the Boundary clock.

  ```
  configure network-clock ptp boundary add {vlan} vlan_name {one-step}
  {master-only | slave-only}
  ```

  ```
  configure network-clock ptp boundary delete {vlan} vlan_name
  ```

- Enable or disable a clock port in the Boundary or Ordinary clock.

  ```
  enable network-clock ptp [boundary | ordinary] {{vlan} vlan_name}
  ```

  ```
  disable network-clock ptp [boundary | ordinary] {{vlan} vlan_name}
  ```

  Note
  Note: A minimum two L3 VLANs (with VLAN port configured) must be configured as Boundary clock ports. Loopback VLAN interface can't be configured as a Boundary clock instance.

- Display the clock ports added to the Boundary or Ordinary clock.

  ```
  show network-clock ptp ordinary [{vlan} vlan_name | vlan all]
  ```

  ```
  show network-clock ptp boundary [{vlan} vlan_name | vlan all]
  ```

  For Ordinary clocks, only unicast master entries can be added on the slave port.

  The query interval for unicast announce messages from the slave port is specified in log base 2.

- Add or remove the unicast master entries on a slave port of the Ordinary clock.

  ```
  configure network-clock ptp ordinary add unicast-master ipv4_address
  {query-interval seconds_log_base_2} {vlan} vlan_name
  ```

  ```
  configure network-clock ptp ordinary delete unicast-master
  ipv4_address {vlan} vlan_name
  ```

  The unicast master entries can be added to the slave port of the Boundary clock. The Boundary clock also support addition of unicast master entries on the port of type 'master or slave'.

- Add or remove unicast master entries on the port of the Boundary clock.

  ```
  configure network-clock ptp boundary add unicast-master ipv4_address
  {query-interval vlan_name} {vlan} vlan_name
  ```

  ```
  configure network-clock ptp boundary delete unicast-master
  ipv4_address {vlan} vlan_name
  ```

- Display the unicast-master entries added to the Boundary or Ordinary clock port.

  ```
  show network-clock ptp boundary unicast-master [{vlan} vlan_name |
  vlan all]
  ```

  ```
  show network-clock ptp ordinary unicast-master [{vlan} vlan_name |
  vlan all]
  ```

- Display the PTP message counters for the peers added to Boundary or Ordinary clock port.

  ```
  show network-clock ptp boundary vlan [vlan_name {{ipv4_address}
  [unicast-master]} | all] counters
  ```

  ```
  show network-clock ptp ordinary vlan [vlan_name {{ipv4_address}
  [unicast-master]} | all] counters
  ```

- Clear the PTP message counters for the peers added to Boundary or Ordinary clock port.

  ```
  clear network-clock ptp boundary vlan [vlan_name {ipv4_address
  [unicast-master]} | all] counters
  ```

  ```
  clear network-clock ptp ordinary vlan [vlan_name {ipv4_address
  [unicast-master]} | all] counters
  ```

- The following properties can be configured on the clock ports added to the Boundary and Ordinary clocks:

  a. Sync message interval:

  ```
  configure network-clock ptp [boundary | ordinary] sync-interval
  seconds_log_base_2 {vlan} vlan_name
  ```

  b. DelayReq message interval:

  ```
  configure network-clock ptp [boundary | ordinary} delay-request-
  interval seconds_log_base_2 {vlan} vlan_name
  ```

  c. Announce message interval:

  ```
  configure network-clock ptp [boundary | ordinary} announce interval
  seconds_log_base_2 {vlan} vlan_name
  ```

  d. Announce message timeout period:

  ```
  configure network-clock ptp [boundary | ordinary] announce timeout
  seconds_log_base_2 {vlan} vlan_name
  ```

PTP Clock Recovery State

- To configure the PTP clock recovery state (servo state) in the system, use the following command:

  ```
  show network-clock ptp
  ```

  The clock recovery using PTP event messages undergoes the following servo state changes:

  - Servo Not Ready - The servo is not yet ready to track the master clock.
  - Servo Ready to Correct - The is ready to track and requests a clock jump to immediately correct the estimated offset.
  - Servo Locked - The servo is tracking the master clock.

PTP Configuration Example

Figure 8 shows a sample configuration using the ExtremeSwitching switches as a transparent clock, a boundary clock, and ordinary clock slaves.

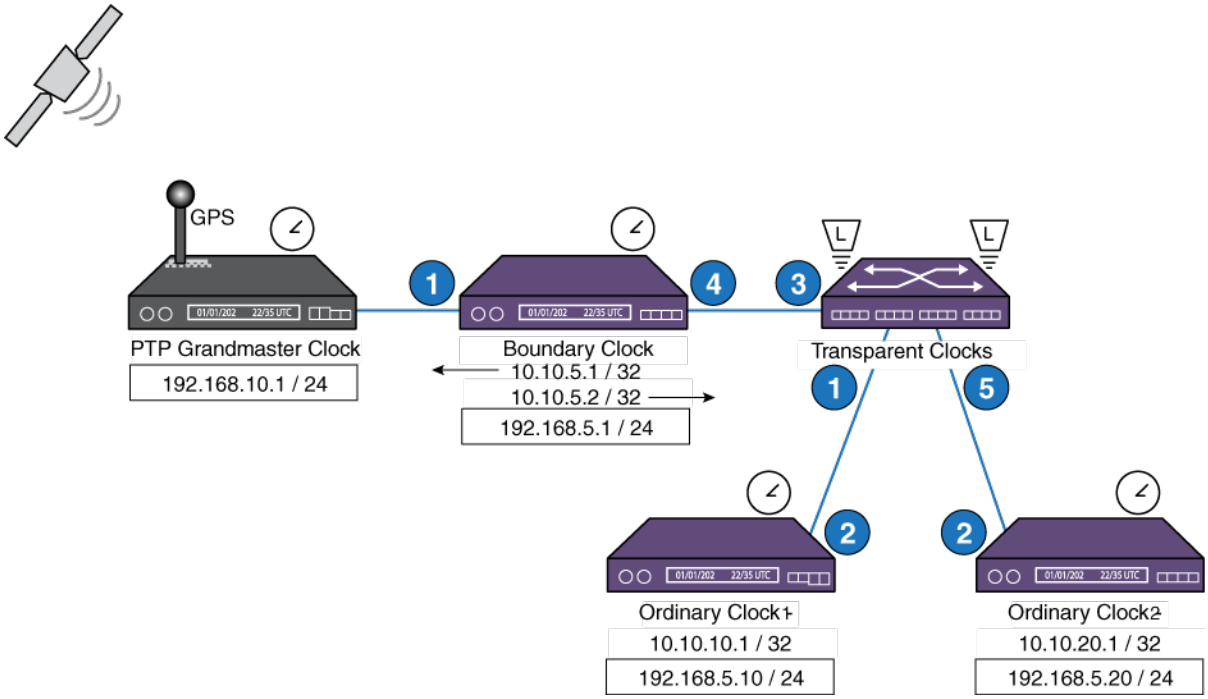


Figure 8: PTP 1588v2 Configuration Example

The IP address of the grandmaster and the IP address of the clock ports in each of the boundary/ordinary clocks are shown in the topology. The IP addresses that are not enclosed in the boxes are assigned to the clock ports added to boundary/ordinary clocks. The transparent clock node can be configured as L2 or L3. In the configuration example below, the transparent clock node is configured as L2.

Note
The grandmaster clock should be reachable from the boundary clock and vice versa. Similarly the ordinary clocks should be reachable from the boundary clock and vice versa. The configuration example below does not consider the provisioning methods used to achieve reachability between the switches, and only limits to the PTP 1588v2 and its associated configuration.

End-to-End Transparent Clock Configuration

In this example, the transparent clock is configured as an L2 switch to transit the PTP event stream between boundary and ordinary clocks.

```
create vlan ptp_tc
configure vlan ptp_tc tag 100
configure vlan ptp_tc add port 1,3,5 tagged
create network-clock ptp end-to-end-transparent
```

```
configure network-clock ptp end-to-end-transparent add ports 1,3,5 one-step
enable network-clock ptp end-to-end-transparent ports 1,3,5
```

Note
The transparent clocks accumulate the residence times on 1-step event messages by performing timestamp in ingress PHYs and in egress PHYs. For proper transparent clock operation, you must ensure in the configuration that the PTP events stream ingress and egress through physical ports that are PTP capable.

Ordinary Clock Slave Configuration

The ordinary clock node is configured to synchronize with the boundary clock node.

The master clock port's IP address in the boundary clock node is added as "unicast-master" in the ordinary clock node.

Note
For PTP event messages originating from ordinary clocks (such as DelayReq), the ingress timestamp for updating the CorrectionField is done in the switch. So you must enable the End-to-End Transparent clock on all the egress ports. Ensure that you do not include the non-PTP capable ports in the configuration of possible egress ports through which the boundary is reachable.

Ordinary Clock Slave Configuration (Node-1)

```
create vlan ptp_slave
configure vlan ptp_slave tag 100
configure vlan ptp_slave add port 2 tagged
configure vlan ptp_slave ipaddress 192.168.5.20/24
enable ipforwarding ptp_slave

create network-clock ptp ordinary
configure network-clock ptp ordinary add vlan ptp_slave  one-step slave-only
configure network-clock ptp ordinary add unicast-master 192.168.5.100 ptp_slave
enable network-clock ptp ordinary
```

Ordinary Clock Slave Configuration (Node-2)

```
create vlan ptp_slave
configure vlan ptp_slave tag 100
configure vlan ptp_slave add port 2 tagged
configure vlan ptp_slave ipaddress 192.168.5.10/24
enable ipforwarding ptp_slave

create network-clock ptp ordinary
configure network-clock ptp ordinary add vlan ptp_slave  one-step slave-only
configure network-clock ptp ordinary add unicast-master 192.168.5.100 ptp_slave
enable network-clock ptp ordinary
```

Boundary Clock Configuration

The boundary clock node is configured to synchronize with the grandmaster node.

The grandmaster clock's IP address is added as "unicast-master" in the boundary clock node. The ptp_gm *VLAN*'s configuration depends on the grandmaster for properties such as tag, or IP.

Note
For boundary clocks, the End-to-End Transparent clock configuration must be applied on the egress ports through with the grandmaster and the ordinary/boundary clocks are reachable.

```
create vlan ptp_gm
configure vlan ptp_gm tag 40
configure vlan ptp_gm add port 1 untagged
configure vlan ptp_gm ipaddress 192.168.10.5/24
enable ipforwarding ptp_gm

create vlan ptp_slaves
configure vlan ptp_slaves tag 100
configure vlan ptp_slaves add port 4 tagged
configure vlan ptp_slaves ipaddress 192.168.5.1/24
enable ipforwarding ptp_slaves

create network-clock ptp boundary
configure network-clock ptp boundary add vlan ptp_gm one-step slave-only
configure network-clock ptp boundary add vlan ptp-slaves one-step master-only
configure network-clock ptp boundary add unicast-master 192.168.10.1 ptp-gm
enable network-clock ptp boundary
```

Note
Loopback IP interface is not added as a PTP clock port. A minimum of two clock ports must be configured for Boundary clock operation. A maximum of 32 PTP clock ports can be configured in Boundary clock.

PTP Profile Configuration Enhancement

Version 33.6.1 adds the ability to specify a PTP profile when creating a PTP domain. When a profile is specified, VLANs subsequently added to the PTP domain automatically inherit the profile's default timing intervals (sync interval, delay request interval, and announce interval) rather than system defaults.

Supported PTP Profiles

The following PTP profiles are supported on 7520 and 7720 platforms:

- **aes67-2018** - AES67-2018 profile (domain 0)
- **g8275.1** - G.8275.1 profile (domain 24)
- **g8275.2** - G.8275.2 profile (domain 44)
- **g8265.1** - G.8265.1 profile (domain 1)
- **st-2059-2** - ST-2059-2 profile (domain 127)

Each profile includes predefined default values for sync interval, delay request interval, and announce interval that conform to the profile's specifications.

Creating a PTP Domain with Profile

To create a PTP domain with a specific profile, use the following command syntax:

```
create network-clock ptp [boundary | ordinary] profile [g8265.1 | g8275.1 | g8275.2 |
aes67-2018 | st-2059-2]
```

When you specify a profile during domain creation, the domain is automatically configured with the appropriate domain number for that profile, and all timing intervals are set according to the profile's defaults.

Example creating a boundary clock with ST-2059-2 profile:

```
create network-clock ptp boundary profile st-2059-2
```

This command creates a PTP boundary clock using domain 127 with ST-2059-2 profile default timing intervals.

Viewing PTP Profile Configuration

To view the configured PTP profile, use the **show network-clock ptp** command. The output displays the profile name in the header:

```
Mode        :  Boundary Clock
Profile     :  st-2059-2
State       :  Enabled
Instance    :  0
================================================================================
                  Default Data Set
================================================================================
      Two Step Flag                    :  FALSE
      Clock Identity                   :  d8:84:66:ff:fe:f6:ec:00
      Number Of Ports                  :  32
      Clock Class                      :  248
      Clock Accuracy                   :  Within 2.5 us
      Offset Scaled Log Variance       :  1
      Priority1                        :  128
      Priority2                        :  128
      Domain Number                    :  127
      Slave Only Flag                  :  FALSE
```

Profile Benefits

Using PTP profiles simplifies configuration by:

- Automatically setting appropriate domain numbers for each profile
- Configuring timing intervals according to profile specifications
- Reducing manual configuration steps when adding VLANs to the domain
- Ensuring compliance with profile-specific requirements

Changing PTP Profiles

To change the profile of an existing PTP domain, you must delete the existing domain and recreate it with the new profile:

```
delete network-clock ptp
create network-clock ptp boundary profile g8275.1
```

Note
Deleting a PTP domain removes all associated configuration. Ensure you document your VLAN and port assignments before deleting the domain.

Running Cable Diagnostics on Ports

You can run cable diagnostics on copper ports to identify physical issues such as cable breaks, shorts, or crosstalk. This feature uses Time Domain Reflectometry (TDR) to test each twisted pair in the cable and report its status.

Command Syntax

`run diagnostics cable ports [port_list | all]`

- Use *`port_list`* to specify one or more copper ports.
- Use all to run diagnostics on all active copper ports.

Note
Diagnostics are only supported on copper ports. They are not available on fiber, stacking, or extended BPE ports. Running diagnostics may temporarily disrupt traffic on active ports.

Interpreting Results

Each twisted pair may report one of the following states:

- `Ok`: No fault detected.
- `Open`: Pair is not terminated or has a physical break.
- `Short`: Conductors are shorted together.
- `Crosstalk`: Interference detected between pairs.

An `Open` state may be expected if the cable is not connected at the far end.

For Gigabit Ethernet (1000BASE-T), all four pairs should report `Ok`. For Fast Ethernet (100BASE-TX), only pairs A and B are used; C and D may show `Open` and still be valid.

For more details and examples, see "Cable Diagnostics" in the **Troubleshooting** chapter.

DWDM Optics Support

DWDM stands for "dense wavelength division multiplexing." Extreme Networks DWDM optics can be tuned to transmit on different "channels" (wavelengths), as needed.

Configuring DWDM

- To configure DWDM specific channels on the port(s), use the following command:

  `configure port **all** | *port_list* **dwdm channel** *channel_number*`

- To configure the DWDM default channel 21 on the port(s), use the following command:

  `configure port **all** | *port_list* **dwdm channel none**`

Displaying DWDM

- To display DWDM configuration, use the following command

  `show ports {**mgmt** | *port_list* | **tag** *tag*} **configuration** {**no-refresh**}`

  `show port {**mgmt** | *port_list* | **tag** *tag*} **information** {**detail**}`

- To display the channel scheme for mapping the DWDM wavelengths, use the following command

  `show dwdm channel-map { *channel_first* { **-** *channel_last* } } {**port** *port_num*}`

Configuring Digital Diagnostic Monitoring Interface (DDMI)

DDMI provides critical system information about the installed optical modules. DDMI is enabled by default.

To enable or disable DDMI, use the following command:

`configure ports [*port_list* | **all**] **ddmi** [**on** | **off**]`

To view DDMI information, use the following commands:

`show ports {*port_list* | **tag** *tag*} **transceiver information**`

`show ports {*port_list* | **tag** *tag*} **transceiver information detail**`

Jumbo Frames

Jumbo frames are Ethernet frames that are larger than 1522 bytes, including four bytes used for the cyclic redundancy check (CRC).

Extreme products support switching and routing of jumbo frames at wire-speed on all ports. The configuration for jumbo frames is saved across reboots of the switch.

Jumbo frames are used between endstations that support larger frame sizes for more efficient transfers of bulk data. Both endstations involved in the transfer must be capable of supporting jumbo frames. The switch only performs IP fragmentation, or participates in maximum transmission unit (MTU) negotiation on behalf of devices that support jumbo frames.

Guidelines for Jumbo Frames

You need jumbo frames when running the Extreme Networks VMAN implementation. With SummitStack or ExtremeSwitching series switches, you can enable and disable jumbo frames on individual ports before configuring VMANs. For more information on configuring VMANs, refer to VLANs on page 2213.

Jumbo frame support is available on ExtremeSwitching series switches that are operating in a SummitStack. For information about displaying jumbo frame status, see Displaying Port Information on page 331.

Enabling Jumbo Frames

Note
Some network interface cards (NICs) have a configured maximum MTU size that does not include the additional 4 bytes of CRC. Ensure that the NIC maximum MTU size is at or below the maximum MTU size configured on the switch. Frames that are larger than the MTU size configured on the switch are dropped at the ingress port.

To enable jumbo frame support, enable jumbo frames on the desired ports:

- To set the maximum jumbo frame size, use the following command:
  `configure jumbo-frame-size framesize`

  The jumbo frame size range is 1,523 to 9,216. This value describes the maximum size of the frame in transit (on the wire), and includes 4 bytes of CRC.

  Note
  For Tagged Packets, the maximum frame size supported is 9,220 bytes, which includes 4 bytes of cyclic redundancy check (CRC), plus another 4 bytes.

- To set the MTU size for the *VLAN*, use the following command:
  `configure ip-mtu mtu vlan vlan_name`
- Next, enable support on the physical ports that will carry jumbo frames using the following command:
  `enable jumbo-frame ports [all | port_list]`

  You can enable or disable jumbo frames for individual ports before configuring the VMANs.

Path MTU Discovery

Using path MTU discovery, a source host assumes that the path MTU is the MTU of the first hop (which is known). The host sends all datagrams on that path with the “don’t fragment” (DF) bit set which restricts fragmentation. If any of the datagrams must be fragmented by an Extreme switch along the path, the Extreme switch discards the datagrams and returns an *ICMP (Internet Control Message Protocol)* Destination Unreachable message to the sending host, with a code meaning "fragmentation needed and DF set." When the source host receives the message (sometimes called

a “Datagram Too Big” message), the source host reduces its assumed path MTU and retransmits the datagrams.

The path MTU discovery process ends when one of the following is true:

- The source host sets the path MTU low enough that its datagrams can be delivered without fragmentation.
- The source host does not set the DF bit in the datagram headers.

If it is willing to have datagrams fragmented, a source host can choose not to set the DF bit in datagram headers. Normally, the host continues to set DF in all datagrams, so that if the route changes and the new path MTU is lower, the host can perform path MTU discovery again.

IP Fragmentation with Jumbo Frames

ExtremeSwitching series switches support fragmentation of IP packets. The above support is included whether or not the switches are present in a SummitStack.

ExtremeXOS supports the fragmenting of IP packets. If an IP packet originates in a local network that allows large packets and those packets traverse a network that limits packets to a smaller size, the packets are fragmented instead of discarded.

This feature is designed to be used in conjunction with jumbo frames. Frames that are fragmented are not processed at wire-speed within the switch fabric.

Note
Jumbo frame-to-jumbo frame fragmentation is not supported. Only jumbo frame-to-normal frame fragmentation is supported.

1. Enable jumbo frames on the incoming port.
2. Add the port to a *VLAN*.
3. Assign an IP address to the VLAN.
4. Enable ipforwarding on the VLAN.
5. Set the MTU size for the VLAN.

   `configure ip-mtu` *`mtu`* **`vlan`** *`vlan_name`*

   The ip-mtu value ranges between 1500 and 9194, with 1500 the default. However, CLI will allow the maximum limit up to 9216 considering port configuration such as tagging which influences L2 Header size. But the values greater than 9194 may lead to packet loss and hence not recommended.

   Note
   To set the MTU size greater than 1500, all ports in the VLAN must have jumbo frames enabled.

Link Aggregation on the Switch

The link aggregation (also known as load sharing) feature allows you to increase bandwidth and availability by using a group of ports to carry traffic in parallel between switches.

Load sharing, link aggregation, and trunking are terms that have been used interchangeably in Extreme Networks documentation to refer to the same feature, which allows multiple physical ports to be aggregated into one logical port, or *LAG (Link Aggregation Group)*. Refer to IEEE 802.1AX for more information on this feature. The advantages to link aggregation include an increase in bandwidth and link redundancy.

Link Aggregation Overview

Link aggregation, or load sharing, is disabled by default. Load sharing allows the switch to use multiple ports as a single logical port, or *LAG*.

Note
All ports in a LAG must be running at the same speed and duplex setting. Each port can belong to only one LAG.

For example, VLANs see the LAG as a single logical port. And, although you can only reference the master port of a LAG to a *STPD (Spanning Tree Domain)*, all the ports of the LAG actually belong to the specified STPD.

If a port in a load-sharing group (or LAG) fails, traffic is redistributed to the remaining ports in the LAG. If the failed port becomes active again, traffic is redistributed to include that port.

Note
Load sharing must be enabled on both ends of the link, or a network loop may result.

Link aggregation is most useful when:

- The egress bandwidth of traffic exceeds the capacity of a single link.
- Multiple links are used for network resiliency.

In both situations, the aggregation of separate physical links into a single logical link multiplies total link bandwidth in addition to providing resiliency against individual link failures.

The software supports control protocols across the LAGs, both static and dynamic. If you add the protocols (for example, *EAPS*, *ESRP (Extreme Standby Router Protocol)*, and so forth) to the port and then create a LAG on that port, you may experience a slight interruption in the protocol operation. To seamlessly add or delete bandwidth when running control protocols, we recommend that you create a LAG consisting of only one port. Then add your protocols to that port and add other ports as needed.

Configurable Per Slot LAG Member Port Distribution

Rather than distribute to all active members in a *LAG*, you have two options to specify a subset of the active member ports as eligible for distribution on a per slot basis: “local slot distribution” and “distribution port lists”. The specific choice of configuration is described in the command line syntax as a distribution-mode. The choice of

distribution mode is configurable per LAG. You may dynamically switch between distribution modes using the `configure sharing` **`distribution-mode`** command.

Local Slot Distribution

The “local-slot” distribution mode restricts distribution of unicast packets to the active LAG members on the same slot where the packet was received. If no active LAG members are present on the slot where the packet was received, all active LAG member ports are included in the distribution algorithm. The “local-slot” distribution mode is useful for reducing the fabric bandwidth load of a switch. Reducing fabric bandwidth may be especially important for a SummitStack, which has significantly less fabric (inter-slot) bandwidth available in comparison to chassis switches. In many SummitStack hardware configurations, the “local-slot” distribution mode may reduce the switching latency of some flows distributed to a LAG.

Distribution Port Lists

The “port-lists” distribution mode configures one or more LAG member ports to be eligible for unicast LAG distribution on each slot in a switch. If a slot does not have a distribution port list configured or if none of the configured member ports is active in the LAG, all active member ports are eligible for unicast distribution. The use of the “port-lists” distribution mode should be taken into consideration when adding ports to a LAG with the configure sharing command. Any newly added port on a LAG is not available for unicast distribution unless it is also added to the distribution port list of at least one slot.

Limitations

The distribution modes affect only the distribution of known unicast packets on a LAG. Non-unicast packets are distributed among all active members of a LAG.

Link Aggregation and Software-Controlled Redundant Ports

If you are configuring software-controlled redundant ports and link aggregation together, the following rules apply:

- You must unconfigure the software-controlled redundant ports before either configuring or unconfiguring load sharing.
- The entire *LAG* must go down before the software-controlled redundant port takes effect.

Dynamic Versus Static Load Sharing

ExtremeXOS software supports two broad categories of load sharing, or link aggregation.

Dynamic Load Sharing

Dynamic load sharing includes the Link Aggregation Control Protocol (LACP) and Health Check Link Aggregation. The Link Aggregation Control Protocol is used to dynamically determine if link aggregation is possible and then to automatically configure the aggregation. LACP is part of the IEEE 802.1AX standard and allows the

switch to dynamically reconfigure the *LAG*s. The LAG is enabled only when LACP detects that the remote device is also using LACP and is able to join the LAG. Health Check Link Aggregation is used to create a link aggregation group that monitors a particular TCP/IP address and TCP port.

Static Load Sharing

Static load sharing is a grouping of ports specifically configured to load share. The switch ports at each end must be specifically configured as part of a load-sharing group.

Note
The platform-related load sharing algorithms apply to LACP (as well as static load sharing).

Load-Sharing Algorithms

Load-sharing, or link aggregation, algorithms select an egress link for each packet forwarded to egress the *LAG*.

The ExtremeXOS software supports the following types of load sharing algorithms:

- Address based—The egress link is chosen based on packet contents (see Address-Based Load Sharing on page 268).
- Port-based—The egress link is chosen based on the key assigned to the ingress port (see Port-Based Load Sharing on page 271).

ExtremeXOS provides multiple addressed-based algorithms. For some types of traffic, the algorithm is fixed and cannot be changed. For other types of traffic, you can configure an algorithm. Algorithm selection is not intended for use in predictive traffic engineering.

Note
Always reference the master logical port of the load-sharing group when configuring or viewing *VLAN*s. VLANs configured to use other ports in the LAG will have those ports deleted from the VLAN when link aggregation is enabled.

Address-Based Load Sharing

All ExtremeXOS platforms support address-based load sharing.

The following are the types of traffic to which addressed-based algorithms apply and the traffic components used to select egress links:

- IPv4 and IPv6 packets—Load sharing is based on the configured options supported on each platform:
  - L2 algorithm—Layer 2 source and destination MAC addresses. Available on SummitStack and all ExtremeSwitching series switches.
  - L3 algorithm—Layer 3 source and destination IP addresses. Available on ExtremeSwitching series switches and SummitStack.

    - L3_L4 algorithm—Layer 3 and Layer 4, the combined source and destination IP addresses and source and destination TCP and UDP port numbers. Available on ExtremeSwitching series switches.
- Non-IP traffic—The source and destination MAC addresses.

Link Aggregation Standard and Custom Algorithms

All ExtremeSwitching switches and stacks support address-based load sharing.

There are two types of algorithms:

- Standard algorithms.
- Custom algorithms, which offer additional options, including the ability to evaluate IP address information from the inner header of an IP-in-IP or GRE tunnel packet.

Standard Algorithms

The following are the types of traffic to which standard addressed-based algorithms apply and the traffic components used to select egress links:

- Layer 2 frames, Unknown unicast packet and non-IP traffic—The source and destination MAC addresses.
- IPv4 and IPv6 packets—Load sharing is based on the configured options supported on each platform:
    - L2 algorithm—Layer 2 source and destination MAC addresses.
    - L3 algorithm—Layer 3 source and destination IP addresses.
    - L3_L4 algorithm—Layer 3 and Layer 4, the combined source and destination IP addresses and source and destination TCP and UDP port numbers.
- *MPLS (Multiprotocol Label Switching)* packets—The source and destination MAC addresses.

Custom Algorithms

The following are the types of traffic to which custom addressed-based algorithms apply and the traffic components used to select egress links:

- Non-IP Layer 2—Uses the *VLAN* ID, the source and destination MAC addresses, and the ethertype.
- IPv4 packets—Uses IP address information from an IP header, or for tunneled packets, the custom algorithm always uses the inner header of an IP-in-IP or GRE tunnel packet. The configuration options are:
    - The source and destination IPv4 addresses and Layer 4 port numbers (default)
    - The source IP address only,
    - The destination IP address only
    - The source and destination IP addresses

- IPv6 packets—Uses the source and destination IPv6 addresses and Layer 4 port numbers.
- MPLS packets—Uses the top, second, and reserved labels and the source and destination IP addresses.

Note
In a switch having at least one *LAG* group with custom algorithm, the egress port for unknown unicast packets across all LAG groups in switch will be decided based on Layer 3 source and destination IP address.

The following command allows you to enable load sharing and select either a standard algorithm or specify that you want to use a custom algorithm:

```
enable sharing port grouping port_list {algorithm [address-based {L2 |
L3 | L3_L4 | custom} | }]} {lacp | health-check}
```

If you choose the custom option when you enable load sharing, you can use the following command to select a custom load sharing algorithm:

```
configure sharing address-based custom [ipv4 [source-only | destination-
only | source-and-destination] | hash-algorithm [xor | crc-16 | crc-32
[lower | upper]]]
```

The hash-algorithm option controls how the source information (such as an IP address) is used to select the egress port. The xor hash algorithm guarantees that the same egress port is selected for traffic distribution based on a pair of IP addresses, Layer 4 ports, or both, regardless of which is the source and which is the destination.

Hash polarization can be prevented by choosing different load sharing algorithms on neighboring switches in different layers. This approach may not be viable in some situations. For example, the network may require the capabilities of the “custom” algorithm for all LAGs in the network. Another approach to varying the load sharing algorithm among neighboring switches would be to use different hashing algorithms of the “custom” algorithm on different switches. For example, some switches could be configured to use “crc-32 upper” algorithm and some configured to use the “crc-16” algorithm.

To change the algorithm for a previously created LAG, use the command:

```
configure sharing master_port algorithm [address-based [L2 | L3 | L3_L4
| custom] | port-based].
```

Note
Use of the *ACL (Access Control List)* redirect-port-no-sharing *port* action overrides any load-sharing algorithm hash that is generated based on the lookup results. For more information about this action, see LAG Port Selection on page 406.

Another strategy for preventing hash polarization is to configure different hash seed values that are used by the CRC versions of the custom algorithms on the LAG switches.

With different hash seeds used on neighboring switches, hash polarization can be prevented while maintaining otherwise identical distribution algorithms.

To set a random hash seed value, use the command:

`configure sharing` **`address-based custom hash-seed`** `[`*`seed`* `|` **`switch-mac-address`**`]`

By using the switch-mac-address option for the hash seed value, identical load sharing algorithms may be used on all of the LAGs in the network while automatically preventing hash polarization.

Prior to supporting configuring the hash seed (ExtremeXOS 30.1), the default value of the hash seed was 0x7F2193EA. You can restore the legacy default behavior for the hash seed by explicitly configuring this legacy value.

Port-Based Load Sharing

A port-based load sharing key is used to determine the index of the selected aggregator port where the list of aggregator ports is organized as an array sorted by increasing front panel port number with a zero-based index. The index of the selected aggregator port is equal to the key value module of the number of ports in the aggregator.

Note
This feature is supported on all ExtremeSwitching Universal platforms.

index = key value % N

where N = the number of ports in the aggregator

The resulting behavior is that ports with a key value of 0 distribute to the lowest numbered port in an aggregator, ports with a key value of 1 distribute to the second lowest numbered port in an aggregator, etc.

Example

A port-based load sharing group contains aggregator ports 2,4,6 and 8. If the zero-based load sharing key 7 is assigned to port 1, then traffic received on port 1 and forwarded to the group will be transmitted on port 8 according to the following calculation:

7 (key) modulo 4 (number of ports in the aggregator) = 3 (index), which corresponds to port 8 which has zero-based index 3 in the sorted array of aggregator ports as shown in Table 23:

Table 23: Aggregator Ports Array

Index	Keys	Member port
0	0, 4, 8, 12	2
1	1, 5, 9, 13	4

Table 23: Aggregator Ports Array (continued)

Index	Keys	Member port
2	2, 6, 10, 14	6
3	3, 7, 11, 15	8

Port-based Load Sharing Limitations

When considering the selection of a *LAG* algorithm, even distribution over member ports is usually the goal. Full utilization of the LAG's bandwidth requires even distribution. In the absence of even distribution, a single member port may become oversubscribed while other member ports are undersubscribed resulting in traffic loss when the LAG, viewed as an aggregate of member ports, is undersubscribed. LAGs distribute best when the diversity of flows destined for the LAG is large relative to the number of ports in the aggregator. For example, when many thousands of L2 flows are destined to a LAG using the "L2" algorithm, distribution on the LAG is typically good (even). Since the number of ports which will switch to a LAG is unlikely to be much larger (orders of magnitude) than the number of ports in the aggregator, extra care may be required from a network administrator when configuring and/or provisioning a switch using port-based LAGs.

LAG Member Port Distribution Modes

As of ExtremeXOS 21.1, two different configuration options are available for specifying subsets of active member ports as eligible for distribution. Both of these options specify a subset of the active member ports on a per slot basis. The specific choice of configuration is described in the CLI as a "distribution-mode". The choice of distribution mode is configurable per *LAG*.

The distribution modes below affect only the distribution of known unicast packets on a LAG. Non-unicast packets are distributed among all active members of a LAG.

Local Slot Distribution

The "local-slot" distribution mode restricts distribution of unicast packets to the active LAG members on the same slot where the packet was received. If no active LAG members are present on the slot where the packet was received, all active LAG member ports are included in the distribution algorithm.

The "local-slot" distribution mode may be specified during LAG creation with the `configure sharing` *`port`* `slot` *`slot`* `distribution-list [`*`port_list`* `| add` *`port_list`* `|` **`all`**`]` CLI command. It may also be configured dynamically with the "configure sharing" command. This distribution mode is self-configuring in the sense that no configuration is required other than the specification of the "local-slot" distribution mode. Addition or deletion of LAG member ports via the `configure sharing` *`port`* `slot` *`slot`* `distribution-list [`*`port_list`* `| add` *`port_list`* `|` **`all`**`]` command is automatically handled.

The "local-slot" distribution mode is useful for reducing the fabric bandwidth load of a switch.

In many SummitStack hardware configurations, the “local-slot” distribution mode will minimize the switching latency of packets distributed to a LAG. Note that some modules may not provide the minimum possible latency due to the details of the switch’s fabric connections. Some ExtremeSwitching switches have more than one switching ASIC and “local-slot” distribution may still involve inter-ASIC switching through local fabric links.

Distribution Port Lists

The “port-lists” distribution mode provides the ability to configure one or more LAG member ports to be eligible for unicast LAG distribution on each slot in the switch. If a slot does not have a distribution port list configured or if none of the configured member ports is active in the LAG, then all active member ports are eligible for unicast distribution.

The “port-lists” distribution mode may be specified during LAG creation with the `configure sharing` *`port`* `slot` *`slot`* `distribution-list [`*`port_list`* `| add` *`port_list`* `|` **`all`**`]` CLI command. It may also be configured dynamically with the `configure sharing` *`port`* `slot` *`slot`* `distribution-list [`*`port_list`* `| add` *`port_list`* `|` **`all`**`]`command. The use of the “port-lists” distribution mode should be taken into consideration when adding ports to a LAG with the “configure sharing” CLI command. Any newly added port on a LAG will not be available for unicast distribution unless it is also added to the distribution port list of at least one slot.

LACP

You can run the Link Aggregation Control Protocol (LACP) on Extreme Networks devices. LACP enables dynamic load sharing and hot standby for link aggregation links, in accordance with the IEEE 802.3ad standard. All third-party devices supporting LACP run with Extreme Networks devices.

The addition of LACP provides the following enhancements to static load sharing, or link aggregation:

- Automatic configuration
- Rapid configuration and reconfiguration
- Deterministic behavior
- Low risk of duplication or misordering

After you enable load-sharing, the LACP protocol is enabled by default. You configure dynamic link aggregation by first assigning a primary, or logical, port to the group, or *LAG* and then specifying the other ports you want in the LAG.

LACP, using an automatically generated key, determines which links can aggregate. Each link can belong to only one LAG. LACP determines which links are available. The communicating systems negotiate priority for controlling the actions of the entire trunk (LAG), using LACP, based on the lowest system MAC number. You can override this automatic prioritization by configuring the system priority for each LAG.

After you enable and configure LACP, the system sends PDUs (LACPDUs) on the LAG ports. The LACPDUs inform the remote system of the identity of the sending system,

the automatically generated key of the link, and the desired aggregation capabilities of the link. If a key from a particular system on a given link matches a key from that system on another link, those links are aggregatable. After the remote system exchanges LACPDUs with the LAG, the system determines the status of the ports and whether to send traffic on which ports.

Among those ports deemed aggregatable by LACP, the system uses those ports with the lowest port number as active ports; the remaining ports aggregatable to that LAG are put into standby status. Should an active link fail, the standby ports become active, also according to the lowest port number. (See Configuring LACP on page 283 for the number of active and standby LACP links supported per platform.)

All ports configured in a LAG begin in an unselected state. Based on the LACPDUs exchanged with the remote link, those ports that have a matching key are moved into a selected state. If there is no matching key, the ports in the LAG remain in the unselected state.

However,if more ports in the LAG are selected than the aggregator can handle because of the system hardware, those ports that fall out of the hardware's capability are moved into standby state. The lowest numbered ports are the first to be automatically added to the aggregator; the rest go to standby. As the name implies, these ports are available to join the aggregator if one of the selected ports should go offline.

You can configure the port priority to ensure the order that ports join the aggregator. However, that port must first be added to the LAG before you can configure the LACP settings. Again, if more than one port is configured with the same priority, the lowest-numbered port joins the aggregator first.

After the ports in the LAG move into the selected state, LACP uses the mux portion of the protocol to determine which ports join the aggregator and can collect and distribute traffic. A few seconds after a port is selected, it moves into the mux state of waiting, and then into the mux state of attached. The attached ports then send their own LACP sync messages announcing that they are ready to receive traffic.

The protocol keeps sending and receiving LACPDUs until both sides of the link have echoed back each other's information; the ends of the link are then considered synchronized. After the sync messages match up on each end, that port is moved into the aggregator (into the mux state of collecting-distributing) and is able to collect and distribute traffic.

The protocol then enables the aggregated link for traffic and monitors the status of the links for changes that may require reconfiguration. For example, if one of the links in a LAG goes down and there are standby links in that LAG, LACP automatically moves the standby port into selected mode and that port begins collecting and distributing traffic.

The marker protocol portion of LACP ensures that all traffic on a link has been received in the order in which it was sent and is used when links must be dynamically moved

between aggregation groups. The Extreme Networks LACP implementation responds to marker frames but does not initiate these frames.

Note
Always verify the LACP configuration by issuing the `show ports sharing` command; look for the ports specified as being in the aggregator. You can also display the aggregator count by issuing the `show lacp lag` command.

You can configure additional parameters for the LACP protocol and the system sends certain *SNMP* traps in conjunction with LACP. The system sends a trap when a member port is added to or deleted from an aggregator.

The system now detects and blocks loopbacks; that is, the system does not allow a pair of ports that are in the same LAG but are connected to one another by the same link to select the same aggregator. If a loopback condition exists between two ports, they cannot aggregate. Ports with the same MAC address and the same admin key cannot aggregate; ports with the same MAC address and a different admin key can belong to the same LAG.

The system sends an error message if a LAG port is configured and up but still not attached to the aggregator or in operation within 60 seconds. Use the `show lacp member-port` *port* `detail` command to display the churn on both sides of the link. If the churn value is shown as True in the display, check your LACP configuration. The issue may be either on your end or on the partner link, but you should check your configuration. The display shows as True until the aggregator forms, and then it changes to False.

A LAG port moves to expired and then to the defaulted state when it fails to receive an LACPDU from its partner for a specified time. You can configure this timeout value as long, which is 90 seconds, or short, which is three seconds; the default is long. (In ExtremeXOS 11.3, the timeout value is not configurable and is set as long, or 90 seconds.) Use the `show lacp lag` *group-id* detail command to display the timeout value for the LAG.

There are two LACP activity modes: active and passive. In LACP active mode, the switch periodically sends LACPDUs; in passive mode, the switch sends LACPDUs only when it receives one from the other end of the link. The default is active mode. (In ExtremeXOS 11.3, the mode is not configurable; it is always active mode.) Use the `show lacp lag` *group-id* `detail` command to display the LACP mode for the LAG.

Note
One side of the link must be in active mode in order to pass traffic. If you configure your side in the passive mode, ensure that the partner link is in LACP active mode.

A LAG port moves into a defaulted state after the timeout value expires with no LACPDUs received for the other side of the link. You can configure whether you want this defaulted LAG port removed from the aggregator or added back into the aggregator. If you configure the LAG to remove ports that move into the default state, those ports are removed from the aggregator and the port state is set to Unselected.

The default configuration for defaulted ports is to be removed, or deleted, from the aggregator. (In ExtremeXOS version 11.3, defaulted ports in the LAG are always removed from the aggregator; this is not configurable.)

Note

To force the LACP trunk to behave like a static sharing trunk, use the `configure sharing` *`port`* **`lacp defaulted-state-action`** [**`add`** | **`delete`**] command to add ports to the aggregator.

If you configure the LAG to add the defaulted port into the aggregator, the system takes inventory of the number of ports currently in the aggregator. If there are fewer ports in the aggregator than the maximum number allowed, the system adds the defaulted port to the aggregator (port set to selected and collecting-distributing). If the aggregator has the maximum ports, the system adds the defaulted port to the standby list (port set to standby). Use the `show lacp lag` *`group-id`* {**`detail`**} command to display the defaulted action set for the LAG.

Note

If the defaulted port is assigned to standby, that port automatically has a lower priority than any other port in the LAG (including those already in standby).

LACP Fallback

Preboot Execution Environment (PXE) is a client/server environment that allows workstations to boot from the server before their full operating system is running. PXE images are too small to take advantage of LACP functionality, and therefore it is up to the administrator to statically configure the switch for correct connectivity. This also means that after the full operating is fully running the switch needs to be reconfigured back for LACP. The LACP fallback feature introduced in ExtremeXOS 21.1 allows for this process to be automated.

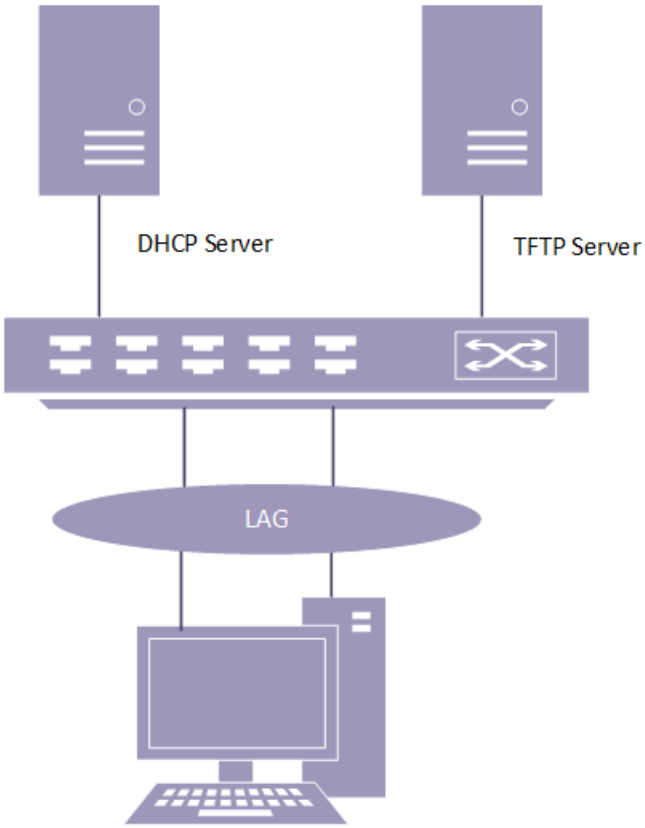


Figure 9: LACP Fallback

The LACP fallback feature allows you to select a single port which will be automatically added to the aggregator if LACP PDUs are not seen on any of the member ports within a specified period of time. If LACP PDUs are exchanged before this timeout expires, then the aggregator is formed using traditional means. If LACP PDUs are not received, then an active port with the lowest priority level is automatically added to the aggregator (enters fallback state). If ports have the same priority value then the lowest port number on the lowest slot number is chosen.

The selected port stays in fallback mode until fallback is disabled or until LACP PDUs start being received on any of the member ports, at which point the old aggregator is removed and a new one is selected based on information propagated in the LACP PDUs. The new fallback port may also be reelected if the existing fallback port changes state; such as port priority change, link bounce, or port enable/disable.

Note
In an environment, fallback port selection occurs only on the LACP master switch.

Link Aggregation Minimum Active Links

The *LAG* Minimum Links feature allows you to configure a value for the minimum number of active links to keep the entire LAG up. For example, for a LAG consisting of 4 ports and the minimum links set to 2, at least 2 links must be up for the LAG to be up. When the LAG falls below 2 active links, the entire LAG is brought down, and

all applications using this LAG will be informed that the LAG is down. Currently, the implicit minimum link value is 1, which means if there is at least 1 link up the entire LAG will stay up.

To configure this ability, use the `configure sharing` *`port`* **`minimum-active`** *`min_links_active`* command.

Both static and LACP LAGs are supported. In the case of static LAG, we check if the number of active physical member port links is greater than or equal to the user-configured minimum link value. If so, the LAG remains up. If the number of active physical member port links falls below the configured minimum link value, the static LAG is brought DOWN and all applications receive a Link-Down message for this LAG. As soon as the number of Active physical member ports equals or exceed the configured minimum link value, the static LAG is brought UP and all applications receive a Link-Up message for this LAG.

In the case of LACP, we keep track of the how many member ports LACP has requested to be added to the LAG. After successfully negotiating with its peer, LACP will send a request to add a member port to the LAG. When the number of member ports LACP has requested to be added to the LAG drops below the configured minimum link value, the LACP LAG will be brought down and all applications will receive a Link-Down message for this LAG. As soon LACP has added enough member ports such that the total member ports equals or exceed the configured minimum link value, the LAG is brought up and all applications receive a Link-Up message for this LAG.

Both static and LACP LAGs can be used with on either the ISC or the MLAG ports. We will verify that the minimum links feature behaves properly for static and LACP LAGs used by MLAG.

Note
For static LAGs, ensure that both ends of the LAG use the same value for minimum links. If the minimum links value differs, one side may see the LAG as down, where the other side may see the LAG as up.

Health Check Link Aggregation

The Health Check *LAG* application allows you to create a link aggregation group where individual member links can monitor a particular TCP/IP address and TCP port.

When connectivity to the TCP/IP address and TCP port fails, the member link is removed from the link aggregation group.

Establishing the status of a TCP connectivity is based on standard TCP socket connections. As long as the switch can establish a TCP connection to the target switch and TCP port, the connection is considered up. The TCP connection will retry based on the configured frequency and miss settings.

A typical use case for this application is when a user wishes to connect each member link to a Security Server to validate traffic. Each member link of the Health Check LAG is connected to an individual Security Server. The LAG is added to a *VLAN* on the same

subnet as the Security Server IP addresses they wish to monitor. Each member port is configured to monitor a particular IP address and TCP port. The Health Check LAG application attempts to do a TCP connect to each IP/TCP port through each member port. The Health Check LAG, by virtue of the sharing algorithm, will load balance traffic across the member links. If a TCP connection cannot be established through the member link, the port is removed from the aggregator and traffic through that particular link is redistributed to the other LAG member links.

Figure 10 displays an example of a Health Check LAG.

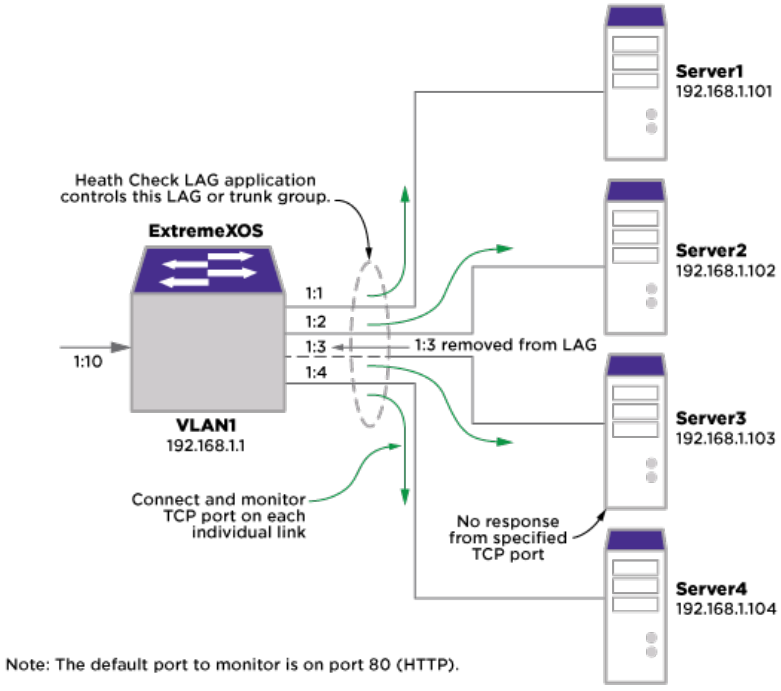


Figure 10: Health Check LAG Example

Guidelines for Load Sharing

Load Sharing Guidelines

For ExtremeXOS standalone switches and SummitStacks the following rules apply to load sharing.

- A static *LAG* can contain up to eight ports.
- An LACP LAG can contain twice the number of ports as a static LAG. The maximum number of selected links is the same as the limit for a static LAG. The remaining links are standby links.
- A Health Check LAG can contain the same number of ports as a static LAG.
- You can configure only the address-based load-sharing algorithm as described in the following sections:
  - Address-Based Load Sharing on page 268

- Link Aggregation Standard and Custom Algorithms on page 269

Note
See Configuring LACP on page 283 for the maximum number of links, selected and standby, per LACP.

Guidelines for the Switches

The following rules apply to load sharing for all Universal platforms:

- A static LAG can contain up to 32 ports when configured to use the L2,L3,L3_L4 or custom algorithm.
- For all the algorithms, LACP LAG can contain up to 64 ports per LAG, which includes up to 32 selected links and 32 standby links.
- A SummitStack consisting entirely of 5520 or 5720 switches can contain up to 64 ports for all algorithms. All platforms have the same stacking limits.
- A SummitStack consisting entirely of 5320 or 5420 switches can contain up to 8 ports for all algorithms. A stack consisting of only 4120 switches or only 4220 switches also can contain up to 8 ports for all algorithms.

Load Sharing Rules and Restrictions for All Switches

Additionally, the following rules apply to load sharing on all switches:

- The ports in the *LAG* do not need to be contiguous. The ports in the LAG can span multiple stacked switches.
- A LAG that spans multiple nodes must use ports that have the same maximum bandwidth capability, with one exception—you can mix media type on 1 Gbps ports.
- On both ingress and egress direction, when you configure an *ACL* to a LAG group, you must configure each of the member ports exclusively.

Enhanced Load Sharing Groups Capacity

Version 33.6 increases the maximum number of load sharing groups (LAGs) on select high-capacity Universal Platforms, enabling larger network deployments with more aggregated links.

Platform Capacity

The maximum number of load sharing groups varies by platform capability:

- **512 LAGs:** High-capacity Universal Platforms (7520, 7720 series)
- **128 LAGs:** All other platforms

The general rule of 8 ports per LAG applies to all platforms regardless of the maximum LAG count. These limits apply to both standalone switches and stack configurations.

Configuring Load Sharing Groups

The syntax for creating and managing load sharing groups remains unchanged. The enhanced capacity is available automatically on supported platforms:

```
enable sharing <port> grouping <port_list> {algorithm [address-based {L2 | L3 | L3_L4 |
custom}]}
```

Example creating a load sharing group:

```
enable sharing 1 grouping 1-4 algorithm address-based L3_L4
```

Stack Considerations

In stack configurations with mixed platform capacities, the system intelligently manages LAG limits to ensure stable operation.

Mixed-Capacity Stack Behavior:

When a node with lower LAG capacity joins a stack where the active LAG count exceeds its supported limit:

- The new node's ports remain disabled to prevent network flooding
- An EMS warning message indicates the LAG capacity mismatch and the number of LAGs that must be removed
- The stack continues to operate with the existing LAGs on nodes that support the current count

Resolving Capacity Mismatches:

To restore full stack operation when a capacity mismatch occurs, choose one of the following approaches:

1. **Reduce LAG count:** Remove LAGs until the total is at or below the lowest capacity in the stack. The new node's ports will become operational once the LAG count is reduced.
2. **Reconfigure stack topology:** Use the **`reboot stacktopology`** command to reinitialize the stack. This clears all LAG configurations and allows you to rebuild your configuration within the new limits.

Port Addition Restrictions:

The system prevents adding ports from lower-capacity nodes to LAGs when the total number of configured LAGs exceeds that node's limit. When attempting to add such ports, an error message indicates the capacity restriction and the number of LAGs that must be removed before the operation can succeed.

Example error message:

```
Cannot add port 2:5 to Load Share Group. Slot 2 supports only 128 Load Share Groups
but system has 256 Load Share Groups configured. Reduce Load Share Groups to 128 or
fewer before adding ports from this slot.
```

Best Practices

- When building stacks, use platforms with similar LAG capacity to avoid operational restrictions
- Document your LAG configuration before adding nodes with different capacities to a stack
- Monitor EMS messages when adding new nodes to stacks with high LAG counts
- Plan LAG deployment to stay within the limits of the lowest-capacity platform in your stack

Viewing Load Sharing Group Information

Use the **show sharing** command to display current load sharing group configuration and port membership. The system does not explicitly show the maximum LAG capacity in command output, but the platform documentation specifies the limits for each platform type.

Configuring Switch Load Sharing

To set up a switch for load sharing, or link aggregation, among ports, you must create a load-sharing group of ports, also known as a *LAG*.

Note
See Guidelines for Load Sharing on page 279 for specific information on load sharing for each specific device.

The first port in the load-sharing group is configured to be the master logical port. This is the reference port used in configuration commands and serves as the LAG group ID. It can be thought of as the logical port representing the entire port group.

All the ports in a load-sharing group must have the same exact configuration, including autonegotiation, duplex setting, *ESRP* host attach or don't-count, and so on. All the ports in a load-sharing group must also be of the same bandwidth class.

Creating and Deleting Load Sharing Groups

To define a load-sharing group, or *LAG*, you assign a group of ports to a single, logical port number.

- To enable or disable a load-sharing group, use the following command

  `enable sharing` *`port`* **`grouping`** *`port_list`* {**`algorithm`** [**`address-based`** {**`L2`** | **`L3`** | **`L3_L4`** | **`custom}`** | }]} {**`lacp`** | **`health-check`**}

  `disable sharing` *`port`*

Note
All ports that are designated for the LAG must be removed from all VLANs prior to configuring the LAG.

Adding and Deleting Ports in a Load-Sharing Group

Ports can be added or deleted dynamically in a load-sharing group, or Load sharing group (*LAG*).

Note
When aggregator membership changes on a LAG, both redistribution of flows, as well as some traffic loss, may be expected.

- To add or delete ports from a load-sharing group, use the following commands:

  `configure sharing` *`port`* **`add ports`** *`port_list`*

```
configure sharing port delete ports port_list
```

Note
See Configuring LACP on page 283 for the maximum number of links, selected and standby, per LACP.

Configuring the Load Sharing Algorithm

For some traffic on selected platforms, you can configure the load sharing algorithm. This is described in Load-Sharing Algorithms on page 268.

- The commands for configuring load sharing algorithms are:

  On SummitStack and all ExtremeSwitching series switches:

  ```
  enable sharing port grouping port_list {algorithm [address-based {L2 | L3 | L3_L4 | custom} | }]} {lacp | health-check}
  ```

  On SummitStack:

  ```
  enable sharing port grouping port_list {algorithm [address-based {L2 | L3 | L3_L4 | custom} | }]} {lacp | health-check}
  ```

  On all platforms:

  ```
  configure sharing address-based custom [ipv4 [L3-and-L4 | source-only | destination-only | source-and-destination] | hash-algorithm [xor | crc-16]]
  ```

Configuring LACP

To configure LACP, you must, again, first create a *LAG*. The first port in the LAG serves as the logical port for the LAG. This is the reference port used in configuration commands. It can be thought of as the logical port representing the entire port group, and it serves as the LAG Group ID.

1. Create a LAG using:

   ```
   enable sharing port grouping port_list {algorithm [address-based {L2 | L3 | L3_L4 | custom} | }]} {lacp | health-check}
   ```

   The port you assign using the first parameter becomes the logical port for the link aggregation group and the LAG Group ID when using LACP. This logical port must also be included in the port list of the grouping itself.
2. If you want to override the default prioritization in LACP for a specified LAG, use:

   ```
   configure sharing port lacp system-priority priority
   ```

   This step is optional; LACP handles prioritization using system MAC addresses.
3. Add or delete ports to the LAG as desired using:

   ```
   configure sharing port add ports port_list
   ```
4. Override the ports selection for joining the LAG by configuring a priority for a port within a LAG by using the command:

   ```
   configure lacp member-port port priority port_priority
   ```

5. Change the expiry timer using:

   `configure sharing` *port* **lacp timeout** [**long** | **short**]

   The default value for the timeout is long, or 90 seconds.
6. Change the activity mode using:

   `configure sharing` *port* **lacp activity-mode** [**active** | **passive**]

   The default value for the activity mode is active.
7. Configure the action the switch takes for defaulted LAG ports.

   `configure sharing` *port* **lacp defaulted-state-action** [**add** | **delete**]

   The default value for defaulted LAG ports is delete the default ports.

Note
Always verify the LACP configuration by issuing the `show ports sharing` command; look for the ports listed as being in the aggregator.

Configuring Health Check Link Aggregation

To configure Health Check link aggregation you must first create a *LAG*. One port in the LAG serves as the logical port for the LAG and is the reference port used in configuration commands.

When you create the LAG, no monitoring is initially configured. The LAG is created in the same way that a static LAG is created and if no monitoring is ever created, this LAG behaves like a static LAG.

1. Create a LAG using:

   `enable sharing` *port* **grouping** *port_list* {**algorithm** [**address-based** {**L2** | **L3** | **L3_L4** | **custom}** | }]} {**lacp** | **health-check**}

   The port you assign using the *port* parameter becomes the logical port for the link aggregation group and the LAG Group ID when using Health Check link aggregation. This logical port must also be included in the port list of the grouping itself.
2. Configure monitoring for each member port using:

   `configure sharing health-check member-port` *port* **add tcp-tracking** *IP Address* {**tcp-port** *TC* **Port frequency** *sec* **misses** *count*}

   If the TCP-port, frequency, or misses are not specified, the defaults described in the Switch Engine v33.6.1 Command References are used.

3. Add the LAG to a *VLAN* whose subnet is the same as the configured tracking IP addresses.

   ```
   configure vlan vlan add port lag port [tagged | untagged]
   ```

   All of the tracking IP addresses must be in the same subnet in which the LAG belongs.

   Note
   VLANs to which Health Check LAG ports are to be added must be configured in loopback mode. This is to prevent the VLAN interface from going down if all ports are removed from the Health Check LAG. In a normal LAG when all ports are removed from the aggregator, the trunk is considered DOWN. As a consequence, if this were the only port in the VLAN, the VLAN interface would be brought DOWN as well. In the Health Check LAG situation, this would cause the TCP monitoring to fail because the L3 vlan interface used by TCP monitoring would no longer send or receive TCP data.

Modifying Configured Health Check LAG

The following commands are used to modify the configured Health Check *LAG*.

1. Delete the monitoring configuration for a member port using the following command:

   ```
   configure sharing health-check member-port port delete tcp-tracking IP Address {tcp-port TC Port}
   ```

2. Enable or disable monitoring for a member port in the Health Check LAG using the following command:

   ```
   configure sharing health-check member-port port [disable | enable] tcp-tracking
   ```

Load-Sharing Examples

This section provides examples of how to define load sharing, or link aggregation, on stand-alone switches, as well has defining dynamic link aggregation.

Load Sharing on a Stand-alone Switch

The following example defines a static load-sharing group that contains ports 9 through 12, and uses the first port in the group as the master logical port 9:

```
enable sharing 9 grouping 9-12
```

In this example, logical port 9 represents physical ports 9 through 12.

When using load sharing, you should always reference the master logical port of the load-sharing group (port 9 in the previous example) when configuring or viewing *VLAN*s; the logical port serves as the *LAG* Group ID. VLANs configured to use other ports in the load-sharing group will have those ports deleted from the VLAN when load sharing becomes enabled.

Cross-Module Load Sharing on SummitStack

The following example defines a static load-sharing group that contains ports 9 through 12 on slot 3, ports 7 through 10 on slot 5, and uses port 7 in the slot 5 group as the primary logical port, or *LAG* Group ID:

```
enable sharing 5:7 grouping 3:9-3:12, 5:7-5:10
```

In this example, logical port 5:7 represents physical ports 3:9 through 3:12 and 5:7 through 5:10.

When using load sharing, you should always reference the LAG Group ID of the load-sharing group (port 5:7 in the previous example) when configuring or viewing *VLAN*s. VLANs configured to use other ports in the load-sharing group will have those ports deleted from the VLAN when load sharing becomes enabled.

Address-based load sharing can also span modules.

Single-Module Load Sharing on SummitStack

The following example defines a static load-sharing, or link aggregation, group that contains ports 9 through 12 on slot 3 and uses the first port as the master logical port 9, or *LAG* group ID:

```
enable sharing 3:9 grouping 3:9-3:12
```

In this example, logical port 3:9 represents physical ports 3:9 through 3:12.

LACP Example

The following configuration example:

- Creates a dynamic *LAG* with the logical port (LAG Group ID) of 10 that contains ports 10 through 12.
- Sets the system priority for that LAG to 3.
- Adds port 5 to the LAG.

```
enable sharing 10 grouping 10-12 lacp
configure sharing 10 lacp system-priority 3
configure sharing 10 add port 5
```

Health Check LAG Example

The following example creates a Health Check *LAG* of 4 ports:

```
create vlan v1
configure v1 ip 192.168.1.1/24
enable sharing 5 grouping 5-8 health-check
enable loopback-mode v1
configure v1 add port 5
configure sharing health-check member-port 5 add track-tcp 192.168.1.101 tcp-port 8080
configure sharing health-check member-port 6 add track-tcp 192.168.1.102 tcp-port 8080
configure sharing health-check member-port 7 add track-tcp 192.168.1.103 tcp-port 8080
configure sharing health-check member-port 8 add track-tcp 192.168.1.104 tcp-port 8080
```

MLAG

MLAG Overview

The feature allows you to combine ports on two switches to form a single logical connection to another network device. The other network device can be either a server or a switch that is separately configured with a regular *LAG* (or appropriate server port teaming) to form the port aggregation.

The following diagram displays the elements in a basic MLAG configuration:

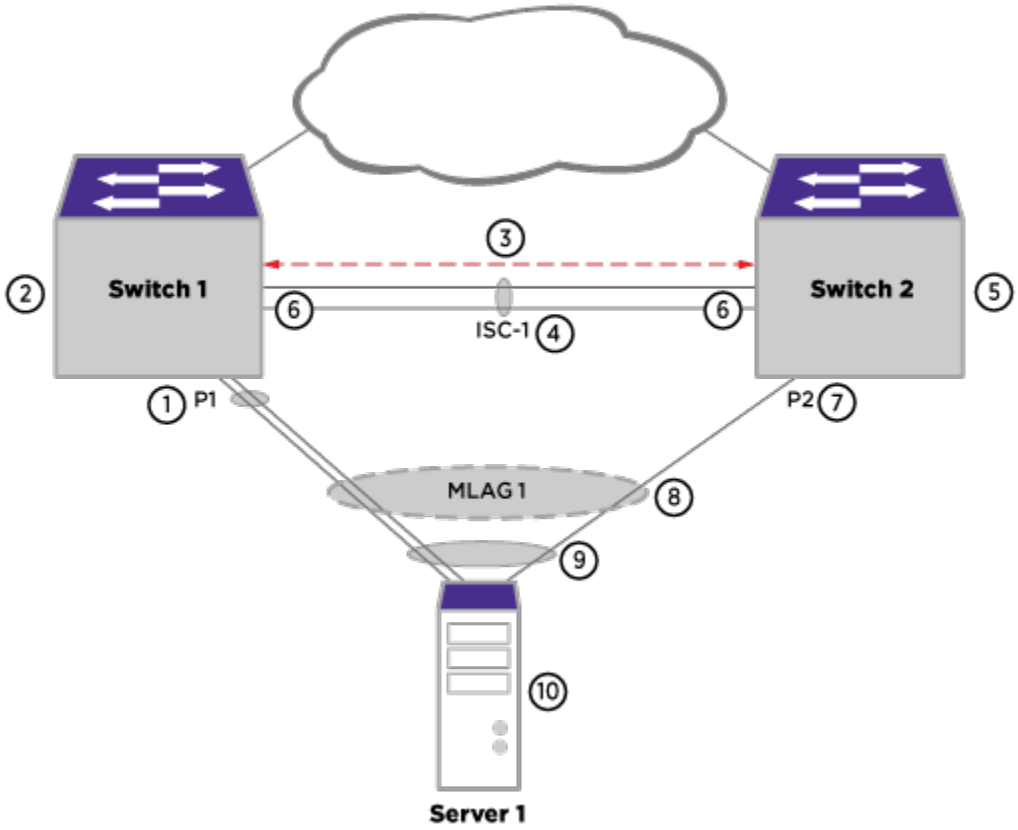


Figure 11: MLAG Elements

1. MLAG port that is a load-shared link. This port is the peer MLAG port for <Switch2:Port P2>.
2. MLAG peer switch for Switch 2.
3. Inter-switch connection (ISC or ISC *VLAN*) has only the ISC port as a member port on both MLAG peers.
4. ISC link that connects MLAG peers.
5. MLAG peer switch for Switch 1.
6. ISC ports.
7. MLAG port that is a non-load-shared link. This port is the peer MLAG port for <Switch1:Port P1>.

8. MLAG group (MLAG-ID 1) that has two member ports (one load-shared and one non-load-shared member).
9. MLAG remote node sees the MLAG ports as regular load-shared link.
10. MLAG remote node - can be a server or a switch.

The operation of this feature requires two ExtremeXOS switches interconnected by an Inter-Switch connection (ISC). The ISC is a normal, directly connected, Ethernet connection and it is recommended that you engineer reliability, redundancy where applicable, and higher bandwidth for the ISC connection. Logically aggregate ports on each of the two switches by assigning MLAG identifiers (MLAG-ID). Ports with the same MLAG-ID are combined to form a single logical network connection. Each MLAG can be comprised of a single link or a LAG on each switch. When an MLAG port is a LAG, the MLAG port state remains up until all ports in the LAG go down.

As long as at least one port in the LAG remains active, the MLAG port state remains active. When an MLAG port (a single port or all ports in a LAG) fails, any associated MAC *FDB (forwarding database)* entries are moved to the ISC, forcing traffic destined to the MLAG to be handled by the MLAG peer switch. Additionally, the MLAG peer switch is notified of the failure and changes its ISC blocking filter to allow transmission to the MLAG peer port. In order to reduce failure convergence time, you can configure MLAG to use ACLs for redirecting traffic via the "fast" convergence-control option.

Each of the two switches maintains the MLAG state for each of the MLAG ports and communicates with each other to learn the MLAG states, MAC FDB entries, and IP multicast entries of the peer MLAG switch.

ISC Blocking Filters

The ISC blocking filters are used to prevent looping and optimize bandwidth utilization.

When at least one peer port is active, the upper layer software initiates a block of traffic that ingresses the ISC port and needs to be forwarded to the local MLAG ports. This is considered to be the steady state condition. In normal steady state operation most network traffic does not traverse the ISC. All unicast packets destined to MLAG ports are sent to the local MLAG port only. However, flood and multicast traffic will traverse the ISC but will be dropped from MLAG peer port transmission by the ISC blocking filter mechanism. The ISC blocking filter matches all Layer 2 traffic received on the ISC and blocks transmission to all MLAG ports that have MLAG peer ports in the active state.

When there are no active MLAG peer ports, the upper layer software initiates an unblocking of traffic that ingresses the ISC port and needs to be forwarded to the local MLAG ports thus providing redundancy. This is considered to be the failed state.

Linkup Isolation

Under certain circumstances, a temporary (less than a second) loop condition exists when an MLAG port becomes operational, but before the remote peer installs the ISC blocking filter (see ISC Blocking Filters on page 288). MLAG linkup isolation addresses this condition by preventing any flood traffic (broadcast, unknown, unicast, etc.) received on a just operational MLAG port from being forwarded to ISC ports until the remote MLAG peer installs the ISC blocking filter.

Inter-Switch Communication

Keep-alive Protocol

peers monitor the health of the ISC using a keep-alive protocol that periodically sends health-check messages. The frequency of these health-check hellos is configurable. When the MLAG switch stops receiving health check messages from the peer, it could be because of the following reasons:

- Failure of the ISC link when the remote peer is still active.
- The remote peer went down.

If the ISC link alone goes down when the remote peer is alive, both the MLAG peers forward the south-bound traffic, resulting in duplication of traffic. However, this does not result in traffic loops. This is because the remote node load shares to both the MLAG peers and does not forward the traffic received on one of the load shared member ports to other member ports of the same load shared group.

Starting in ExtremeXOS 15.5, health check messages can also be exchanged on an alternate path by separate configuration – typically the “Mgmt” *VLAN*. If the peer is alive when the ISC link alone goes down, one of the MLAG peers disables its MLAG ports to prevent duplicate south-bound traffic to the remote node. To reduce the amount of traffic on the alternate path, health check messages are initiated on the alternate path only when the ISC link goes down. When the ISC link is up, no health check messages are exchanged on the alternate path.

When the MLAG switch misses 3 consecutive health check messages from the peer, it declares that the MLAG peer is not reachable on the ISC link. It then starts sending out health check messages on the alternate path to check if the peer is alive. When the first health check message is received from the MLAG peer on the alternate path, it means that the peer is alive. In this scenario, one of the MLAG peers disables its MLAG ports to prevent duplication of south-bound traffic to the remote node.

Note
The MLAG switch having the lower IP address for the alternate path VLAN disables its ports.

When the ISC link comes up and the switch starts receiving health check messages on the ISC control VLAN, the ports that were disabled earlier have to be re-enabled. This action is not performed immediately on the receipt of the first health check message on the ISC control VLAN. Instead the switch waits for 2 seconds before enabling the previously disabled ports. This is done to prevent frequent enabling and disabling of MLAG ports due to a faulty ISC link up event.

MLAG Status Checkpointing

Each switch sends its MLAG peer information about the configuration and status of MLAGs that are currently configured over the ISC link.

This information is checkpointed over a TCP connection that is established between the MLAG peers after the keep-alive protocol has been bootstrapped.

Authentication for Checkpoint Messages

The checkpoint messages exchanged between the MLAG peers over the TCP connection are sent in plain text and can be subjected to spoofing. Starting from ExtremeXOS 15.5 a provision is provided as part of this feature to secure the checkpoint connection against spoofing.

A key for authentication must be configured on both the MLAG peer switches. This key will be used in calculating the authentication digest for the TCP messages. TCP_MD5SIG socket option will be used for authentication and so only *MD5 (Message-Digest algorithm 5)* authentication is supported. The configured key will be used in setting up TCP_MD5SIG option on the checkpoint socket. The same key must be configured on both the MLAG peers. The checkpoint connection will not be established if different keys are configured on the MLAG peer switches.

PIM MLAG Support

ExtremeXOS allows configuring PIM on both peers. PIM adjacencies between each MLAG vlan are established across the ISC link.

Note
PIM MLAG is supported only on default VR and is not supported on user-created VRs.

- The checkpoint PIM state between MLAG peers. This should include all MLAG egresses.
- You can verify that PIM functionality for MLAG is present by issuing the following show command:

  `show pim cache {{`**`detail`**`} | {`**`state-refresh`**`} {`**`mlag-peer-info`**`} {`*`group_addr {source_addr`*`}}}`
- Additionally, the existing show pim cache command displays ingress *VLAN* information for all MLAG peers.

  The output of the command is shown below:

```
Index    Dest Group      Source             InVlan   Origin
          [0001]  226.1.1.1       112.2.2.202 (S)    fifthteen Sparse
       Expires after 203 secs UpstNbr: 51.15.15.2
       RP: 61.2.2.2 via 51.15.15.2 in fifthteen
       Peer Ingress VLAN (ISC 1): 51.15.15.4/24 (Same)
       EgressIfList =  eight(0)(FW)(SM)(I) , five(0)(FW)(SM)(I)
       PrunedIfList =  ten(0)(SM)(AL)
```

Multiple VLAN Registration Protocol (MVRP) over Multi-switch Link Aggregation (MLAG)

This feature adds support for Multiple *VLAN* Registration Protocol (MVRP) on . The objective of running Multiple VLAN Registration Protocol (MVRP) over MLAG is to propagate the VLANs across the MLAG peers and to have the VLAN database synced, so that the remote switches/servers continue to see a single logical connection.

Checkpointing

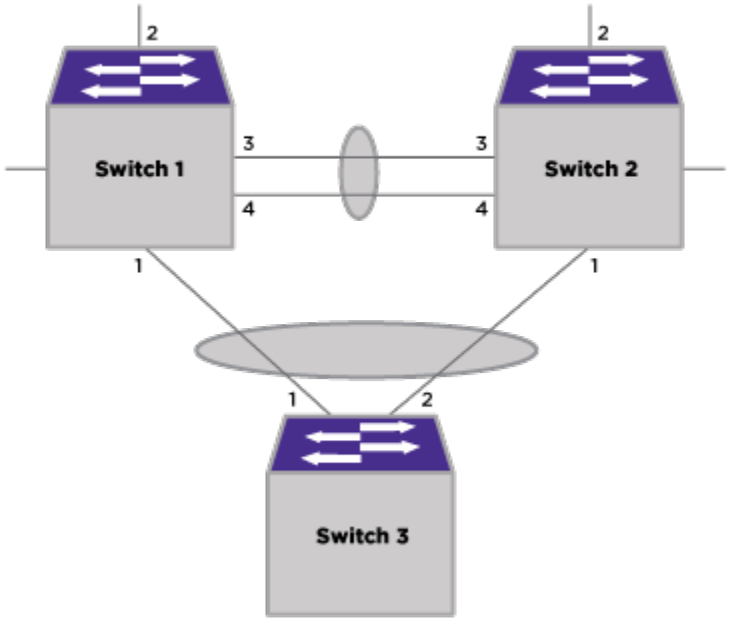


Figure 12: Checkpointing

The primary reason for checkpointing is keeping the MLAG peers synchronized. The VLAN membership status of MLAG ports is maintained similarly through checkpointing.

There are two types of checkpointing that are done:

- PDU checkpointing
- Bulk checkpointing

PDU Checkpointing

Each incoming MRP PDU on a MLAG port is encapsulated and sent to the MLAG peer through the ISC connection. On the MLAG peer, the packet is de-capsulated and processed by the MRP process as if received by the local MLAG port. This ensures that the local MLAG port on the peer is also added to the VLAN. The MVRP propagation happens normally in addition to the PDU checkpointing. In Figure 12 on page 291, Switch 1 and 2 are MLAG peers connected through ISC port (*LAG* ports 3,4). Switch 3 can send the MVRP PDUs to either Switch 1 or Switch 2 based on which MLAG peer is up. Assuming the MVRP PDU is sent to Switch 1, port 1 of Switch 1 is added to the VLAN and the PDU is checkpointed to Switch 2. If checkpointing is not done, port 1 of Switch 2 does not get added to the dynamic VLAN, since the MVRP PDU is not received on that port.

Bulk Checkpointing

When the MLAG peer comes up for the first time, bulk checkpointing of VLAN membership of the MLAG ports occurs. The peer that comes up sends a “pull” message to the active peer. Upon receiving the pull message, MVRP PDU is constructed for each

of the MLAG ports and checkpointed through the ISC connection. Upon receiving the checkpointed PDU, it is decapsulated and processed as if it were received locally by the MRP process. This way the MLAG ports of the peer that came up gets added to VLANs. Bulk checkpointing is also done when MVRP is enabled on the MLAG port.

MVRP Packet Received on MLAG Port Process

The following is the sequence of operations when an MVRP packet is received on a MLAG port.

1. MVRP packet is parsed by MRP process.
2. Upon successful parsing, the MVRP packet is encapsulated by the VSM and sent to the corresponding MLAG peer.
3. The MLAG index is encapsulated along with the packet.
4. In the remote MLAG peer, the packet is de-capsulated.
5. From the MLAG index, the local MLAG port is determined.
6. The MVRP packet is processed as if the PDU has been received by the local MLAG port

Orchestration Mode for Checkpointing MLAG Port Configuration

With , the port configuration for the MLAG member links must be identical on the MLAG peer switches. To reduce the configuration complexity and to minimize the risk of inconsistency, you can use orchestration mode so that any configuration commands are now checkpointed to the MLAG peer switch.

After entering this orchestration mode, like the existing virtual-router mode, the configuration prompt changes indicating that commands issued are within this context:

```
(orchestration bottom) switch model #
```

Note
Enabling orchestration mode on both MLAG peers at same time is not recommended. If orchestration mode is enabled on both MLAG peers and you execute commands on both switches at same time, the switch may abort execution of the command.

For information about the orchestration commands, see Configuring MLAGs on page 299.

Note
If you need to have different settings on each MLAG peer (for example, IP addresses), you need to perform those configuration steps outside of orchestration mode.

Multi-switch Link Aggregation Group (MLAG) Port Reload Delay Timer

On certain platforms, it takes few seconds between the first port and last port to come up. When lower numbered ports are used as Multi-switch Link Aggregation Group (MLAG) ports and higher numbered ports are used as inter-switch connection (ISC) ports, MLAG ports come up first before the ISC ports. In such cases, traffic from the

servers MLAGed to the peers that traverse the ISC can be lost during the duration when the traffic hashes to the MLAG peer while the ISC is still not up.

The MLAG reload delay timer keeps MLAG ports disabled for the configured duration while the switch configuration is loading. This timer is also useful for cases where network-facing Layer 3 protocols, like OSPF, are yet to converge on the node that has just come up. This feature is disabled by default.

Support for More than One MLAG Peer

Beginning in ExtremeXOS 15.5, each switch can support the creation of two MLAG peers. The following topology is now supported:

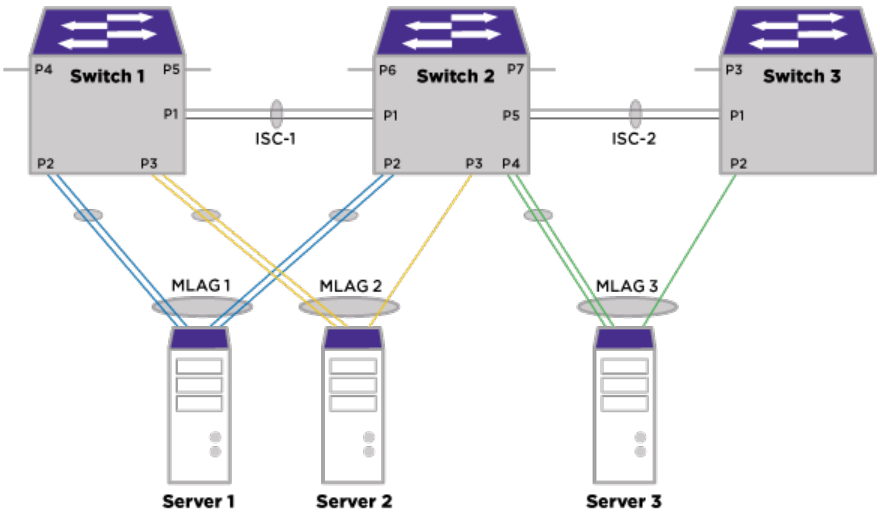


Figure 13: Two MLAG Peer Topology

All basic MLAG functionality and traffic forwarding rules that existed earlier apply with this topology.

Note
A port is an MLAG port only with respect to a particular MLAG peer switch. In the above example, the “green” port on switch-2 is an MLAG port with respect to switch-3 on ISC-2. It is not an MLAG port with respect to switch-1 on ISC-1. Similarly, on switch-2, “blue” and “orange” ports are MLAG ports with respect to switch-1 on ISC-1. They are not MLAG ports with respect to switch-3 on ISC-2.

Traffic Flows

In the steady state when all the ports are enabled, the blocking rule prevents traffic coming from the ISC port to be forwarded to its MLAG ports. So, in steady state, any unknown (unknown unicast/broadcast/multicast) traffic from Switch-1 to Switch-2 over ISC-1 will not be flooded to “blue” and “orange” MLAG ports on switch-2. However, since the “green” MLAG port on switch-2 does not correspond to ISC-1, the traffic from ISC-1 will be forwarded to “green” MLAG port as shown in the following illustration:

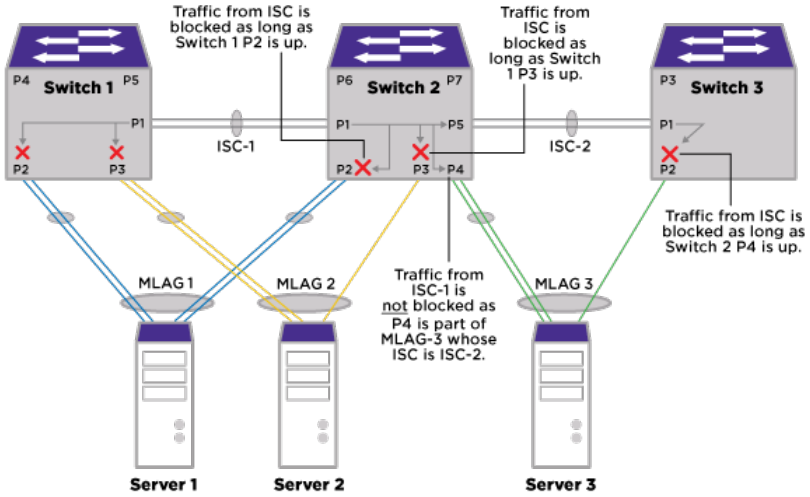


Figure 14: Traffic Flow

In steady state, the unknown traffic from Switch-3 to Switch-2 over ISC-2, will be blocked to “green” MLAG ports. However, this traffic will be sent to “blue” and “orange” MLAG ports.

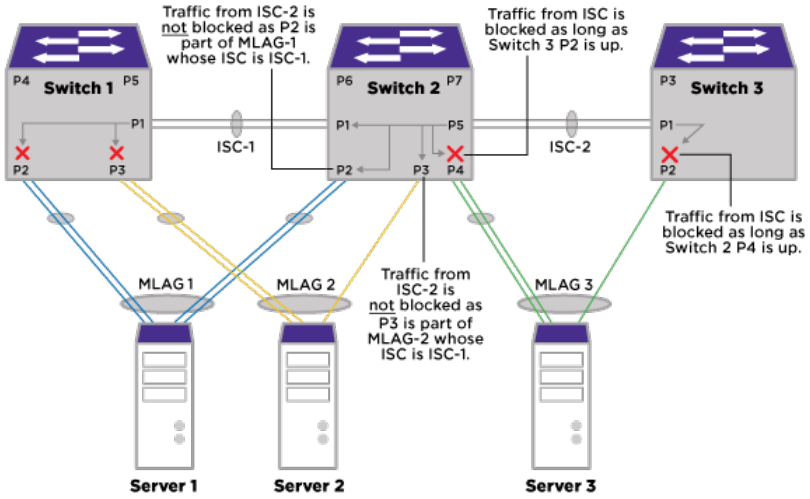


Figure 15: Traffic Flow (Unknown Traffic Switch-3 to Switch-2)

MLAG Peer Port Failure

In the case of an port failure, the blocking rule on the peer MLAG switch will be removed and the traffic coming on the ISC will be forwarded to the corresponding MLAG port.

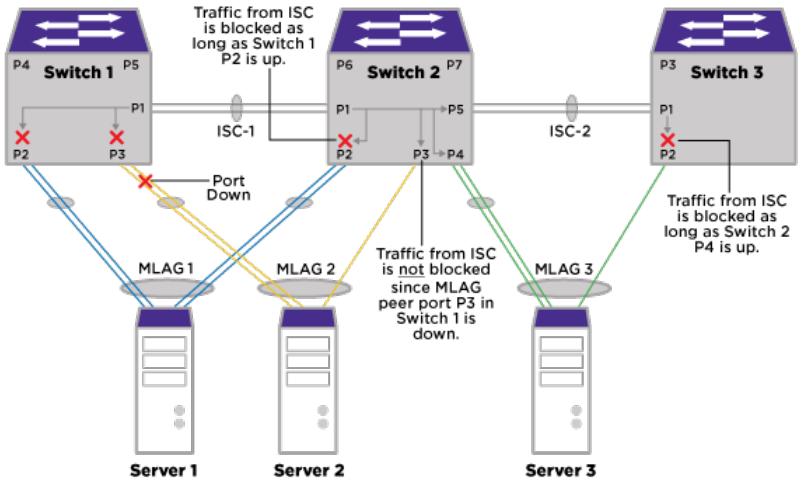


Figure 16: Peer Port Failure

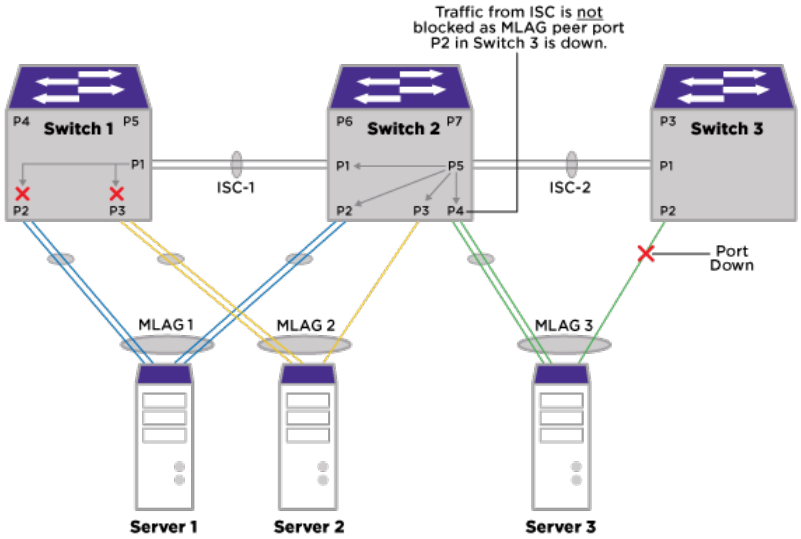


Figure 17: Peer Port Failure (cont.)

MLAG Limitations and Requirements

The feature has the following limitations:

- MLAG peer switches must be of the same platform family. The following MLAG peers are allowed:
  - ExtremeSwitching switches with ExtremeSwitching switches

- SummitStack with SummitStack

Note

In the case of ExtremeSwitching standalone switches, it is strongly recommended that MLAG peer switches be of the same type.

In the case of SummitStack switches, we recommend that the MLAG ports be from slots of similar capability.

- Layer 2 protocols such as *EAPS* or *STP (Spanning Tree Protocol)* will be configured to not allow the blocking of the ISC.
- The number of MLAG ports for each pair of switches is limited to the number of physical ports in the system.
- MLAG peers should run the same version of ExtremeXOS for proper functioning.
- *ESRP* cannot be configured in a MLAG environment with more than one peer.
- The MLAG peers in a multi peer setup cannot be looped however can be extended as a linear daisy chain.

Note

The behavior of ExtremeXOS 30.1 has changed if locally configured IP addresses are used to determine network reachability. For details, see the `ping` command.

MLAG Requirements

The following table shows additional MLAG requirements that are specific to other protocols and features.

Items	Impact
VLAN:Membership	You must add the respective port (or *LAG*) that is part of an MLAG to a VLAN on both MLAG peers. The set of configured VLANs on [Switch1:P1] must be identical to the set of VLANs configured on [Switch2:P2]. You must add the ISC to every VLAN that has an MLAG link as a member port.
VMAN:Membership	The restrictions are the same as those for VLAN Membership.
VLAN:ISC	You must create a Layer 3 VLAN for control communication between MLAG peers. You cannot enable IP forwarding on this VLAN. The ISC is exclusively used for inter-MLAG peer control traffic and should not be provisioned to carry any user data traffic. Customer data traffic however can traverse the ISC port using other user VLANs.
VMAN:ISC	Although not recommended, a VMAN may be configured to carry Inter-MLAG peer traffic,
LAG:Load-Sharing Algorithm	It is recommended but not required that LAGs that form an MLAG be configured to use the same algorithm.
Ports:Flooding	To disable flooding on an MLAG, you must disable flooding on both ports (or LAGs) that form the MLAG.

Items	Impact
Ports:Learning	To disable learning on an MLAG, you must disable learning on both ports (or LAGs) that form the MLAG. Learning is disabled by default on ISC ports.
FDB:Static & Blackhole entries	Configuration must be identical on both MLAG peers for entries that point to an MLAG port.
FDB:Limit learning	Learning limits are applicable to member ports of each peer. The limit on the MLAG is the sum of the configured value on each peer.
FDB:MAC Lockdown	This is supported but needs configuration on both peers. A switch can still receive checkpointed MAC addresses from its peer in the window between executing the lockdown command on both switches.
EAPS	MLAG ports cannot be configured to be EAPS ring ports. Configuration of the ISC port as an EAPS blocked port is disallowed.
STP	You should ensure that the ISC port is never blocked by STP. You should configure in such a way that one of the MLAG peers become root bridge and also configure MLAG peers with backup root option. Loop protect feature should not be enabled on LAG ports on access switches. Enabling this feature on MLAG ports is acceptable. For more information about STP and MLAG, see STP and MLAG on page 1842.
VRRP	VRRP must be enabled on Layer 3 VLANs that have MLAG member ports.
ESRP	MLAG and ISC ports must be added as ESRP host-attach ports.
EDP/*LLDP (Link Layer Discovery Protocol)*	There are no restrictions but the remote end of the MLAG will display different neighbors for different ports in the same LAG.
ELSM	ELSM is not to be configured on MLAG ports at either end of an MLAG.
Software-Redundant Ports	These are not to be configured on MLAG ports at either end of an MLAG.
Mirroring	Mirroring on local ports in an MLAG is supported. Mirroring of MLAG peer ports to a local port is not supported.
Routing Protocols	OSPFV2/OSPFV3 neighborship can be formed across an MLAG.
Multicast:*IGMP (Internet Group Management Protocol)*	All timers related to IGMP must be identical on both the peers.

Items	Impact
Multicast:PIM	PIM should be configured on both the MLAG peers, and the PIM timers must be identical. MLAG functionality must not be enabled on PIM Intermediate routers. It should be enabled only on Last Hop (LHR) and First Hop (FHR) routers. MLAG peer switches S1 and S2 perform Checkpoint PIM for S and G states. This should include all MLAG egresses. To avoid traffic drops due to asserts, do not include ISC port in MLAG egresses if the ingress VLAN includes ISC port, and both the peers have the same ingress for the S, G cache.
Multicast:MVR	MVR should be enabled on only one of the MLAG peer switches. MVR must not be enabled on MLAG VLANs.
Multicast:PIM Snooping	This is not supported.
Multicast:IPv6	There are no restrictions.
CFM	There are no restrictions.
MPLS:General	MPLS cannot be enabled on VLANs having MLAG member ports.
MPLS:VPLS	VPLS must be configured for redundancy using ESRP. The ESRP master VLAN must include the ISC ports and the VPLS service VLAN ports as members. Pseudowires cannot traverse an ISC link. You should not add the ISC port as a member to MPLS VLANs that can be used by LSPs that can carry Layer 2 VPN traffic terminating on MLAG peer switches.
ACLs	It is strongly recommended that configuration be identical across peers on MLAG ports.
QoS	It is strongly recommended that configuration be identical across peers on MLAG ports.
NetLogin	Supported. NetLogin should be enabled across the MLAG ports of both the peers.
VLAN:PVLAN	If an MLAG port is a member of either a subscriber VLAN or a network VLAN, the ISC port needs to be added as a member of the network VLAN. Subscriber VLANs in a private VLAN cannot have overlapping MLAG ports as members. Configuring dedicated loopback ports for subscriber VLANs in a private VLAN that shares an MLAG port causes duplicate traffic to be sent to the remote node.
W-MLAG	Supported.

Configuring MLAGs

This section provides the commands used to configure MLAGs and display information about those configured.

- To create an peer switch association structure, use the following command:
  `create mlag peer peer_name`
- To delete a peer switch, use the following command:
  `delete mlag peer peer_name`
- To associate an MLAG peer structure with an MLAG peer switch IP address, use the following command:
  `configure mlag peer peer_name ipaddress peer_ip_address {vr VR}`
- To unconfigure the association, use the following command:
  `unconfigure mlag peer peer_name ipaddress`
- To configure the time interval between health check hello packets exchanged between MLAG peer switches, use the following command:
  `configure mlag peer peer_name interval msec`
- To unconfigure the time interval setting and reset the interval to the default of 1000 ms, use the following command:
  `unconfigure mlag peer peer_name interval`
- To bind a local port or *LAG* to an MLAG specified with an integer identifier, use the following command:
  `enable mlag port port peer peer_name id identifier`
- To disable a local port or LAG from an MLAG, use the following command:
  `disable mlag port port`
- To enable the current ports associated with the given MLAG ID, use the following command:
  `enable ports [mlag-id mlag_id]`
- To disable the current ports associated with the given MLAG ID, use the following command:
  `disable ports [mlag-id mlag_id]`
- To set a preference for having a fast convergence time or conserving access lists, use the following command:
  `configure mlag ports convergence-control [conserve-access-lists | fast]`

  Note
  Executing the `refresh policy` command with MLAG configuration may result in health check hellos not reaching the CPU. To avoid MLAG peer connectivity disruption, you can either execute the `disable access-list refresh blackhole` command or temporarily increase the peer hello interval to a large value (for instance, 10000 ms) and reset it back once refresh policy is complete.

- To enable or disable MLAG linkup isolation, use the following command:
  `configure mlag ports link-up-isolation [on | off]`

- To start orchestration mode, use the following command:

  `start` **`orchestration mlag`** *`peer_name`*
- To stop orchestration mode, use the following command:

  `stop {`**`orchestration`**`}`
- To configure MLAG reload delay timer, use the following command:

  `configure mlag ports` **`reload-delay`** *`reload-delay`*
- To enable MLAG reload delay timer, use the following command:

  `enable mlag port reload-delay`
- To disable MLAG reload delay timer, use the following command:

  `disable mlag port reload-delay`
- To view the configured MLAG reload delay timer setting, use the following command:

  `show mlag ports {`*`port_list`*`}`
- To rename an existing MLAG peer, use the following command:

  `configure {` **`mlag peer`** `}` *`peer_name`* **`name`** *`new_peer_name`*

Displaying MLAG Information

- To display information about an peer, including MLAG peer switch state, MLAG group count, and health-check statistics, use the following commands:

  `show mlag peer {`*`peer_name`*`}`
- To display each MLAG group, including local port number, local port status, remote MLAG port state, MLAG peer name, MLAG peer status, local port failure count, remote MLAG port failure count, and MLAG peer failure count, use the following command:

  `show mlag ports {`*`port_list`*`}`
- To see if a port is part of an MLAG group, or an ISC port, use the following command:

  `show ports {`**`port_list`** `}` **`information`** `{`**`detail`**`}`

Example of MLAG Configuration

Below is an example of how to configure an . Figure 18 shows a finished MLAG network.

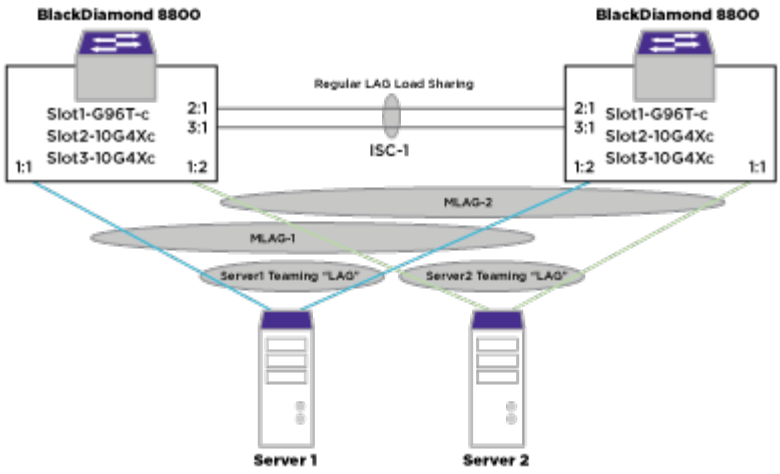


Figure 18: Simple MLAG Configuration

1. Create the Inter-Switch Connection (ISC).

   Description: The ISC provides an out-of-band IP communications path between the two MLAG peer switches to exchange keep-alive packets and to checkpoint various state information between switches.

   On the “Left” BlackDiamond 8800 switch:

   ```
   enable sharing 2:1 group 2:1,3:1
   create vlan isc
   config vlan isc tag 3000
   config vlan isc add port 2:1 tag
   config vlan isc ipaddress 1.1.1.1/24
   ```

   On the “Right” BlackDiamond 8800 switch:

   ```
   enable sharing 2:1 group 2:1,3:1
   create vlan isc
   config vlan isc tag 3000
   config vlan isc add port 2:1 tag
   config vlan isc ipaddress 1.1.1.2/24
   ```

2. Create the MLAG peer and associate the peer switch's IP address.

   Description: By creating an MLAG peer you associate a peer name that can be associated with the peer switch's IP address and other peer configuration properties. The peer is then bound to each individual MLAG port group.

   On the “left” BlackDiamond 8800 switch:

   ```
   create mlag peer "rightBD8K"
   config mlag peer “rightBD8K” ipaddress 1.1.1.2
   ```

   On the “right” BlackDiamond 8800 switch:

   ```
   create mlag peer “leftBD8K”
   config mlag peer "leftBD8K" ipaddress 1.1.1.1
   ```

3. Create the MLAG port groups.

   Tip
   At this point, you could use orchestration mode to allow you to checkpoint all commands issued on the one switch to its MLAG peer. For more information about, MLAG orchestration, see Orchestration Mode for Checkpointing MLAG Port Configuration on page 292.

   Description: Creates an MLAG port group by specifying the local switch's port, the MLAG peer switch, and an "mlag-id" which is used to reference the corresponding port on the MLAG peer switch. The specified local switch's port can be either a single port or a load share master port.

   On the “left” BlackDiamond 8800 switch:

   ```
   enable mlag port 1:1 peer "rightBD8K" id 1
   enable mlag port 1:2 peer "rightBD8K" id 2
   ```

   On the “right” BlackDiamond 8800 switch:

   ```
   enable mlag port 1:2 peer "leftBD8K" id 1
   enable mlag port 1:1 peer "leftBD8K" id 2
   ```

4. Verify MLAG peers and ports are operational.

   Description: After MLAG groups are configured, you can verify the connections via the show mlag peer and show mlag ports commands. Be sure to note the peer status, the Local Link State, and the Remote Link status.

   On the “left” BlackDiamond 8800 switch:

   ```
   BD-8810.5 # show mlag peer
   Multi-switch Link Aggregation Peers:
   MLAG Peer         : leftBD8k
   VLAN              : isc             Virtual Router    : VR-Default
   Local IP Address  : 1.1.1.2         Peer IP Address   : 1.1.1.1
   MLAG ports        : 2               Tx-Interval       : 1000 ms
   Checkpoint Status : Up              Peer Tx-Interval  : 1000 ms
   Rx-Hellos         : 184             Tx-Hellos         : 184
   Rx-Checkpoint Msgs: 12              Tx-Checkpoint Msgs: 12
   Rx-Hello Errors   : 0               Tx-Hello Errors   : 0
   Hello Timeouts    : 1               Checkpoint Errors : 0
   Up Time           : 0d:0h:0m:10s    Peer Conn.Failures: 1

   BD-8810.3 # show mlag ports
   Local
                                                                 Local  Remote
   MLAG        Local   Link     Remote                Peer     Fail   Fail
   Id          Port    State    Link    Peer          Status   Count  Count
   ================================================================================
   1       1:1   A        Up          rightBD8K       Up          0      0
   2       1:2   A        Up          rightBD8K       Up          0      0
   ================================================================================
   Local Link State: A - Active, D - Disabled, R - Ready, NP - Port not present
   Remote Link     : Up - One or more links are active on the remote switch,
   Down - No links are active on the remote switch,
   N/A - The peer has not communicated link state for this MLAG port
   Number of Multi-switch Link Aggregation Groups  : 2
   Convergence control                             : Fast
   ```

On the “right” BlackDiamond 8800 switch:

```
BD-8810.3 # show mlag peer
Multi-switch Link Aggregation Peers:
MLAG Peer         : rightBD8k
VLAN              : isc             Virtual Router    : VR-Default
Local IP Address  : 1.1.1.1         Peer IP Address   : 1.1.1.2
MLAG ports        : 2               Tx-Interval       : 1000 ms
Checkpoint Status : Up              Peer Tx-Interval  : 1000 ms
Rx-Hellos         : 167             Tx-Hellos         : 167
Rx-Checkpoint Msgs: 12              Tx-Checkpoint Msgs: 12
Rx-Hello Errors   : 0               Tx-Hello Errors   : 0
Hello Timeouts    : 1               Checkpoint Errors : 0
Up Time           : 0d:0h:0m:7s     Peer Conn.Failures: 1

BD-8810.5 # show mlag ports
Local                                                 Local   Remote
MLAG            Local   Link     Remote                       Peer    Fail
FailId
                Port    State    Link    Peer                 Status  Count   Count
================================================================================
2       1:1     A        Up              leftBD8K             Up          0       0
1       1:2     A        Up              leftBD8K             Up          0       0
================================================================================
Local Link State: A - Active, D - Disabled, R - Ready, NP - Port not present
Remote Link     : Up - One or more links are active on the remote switch,
Down - No links are active on the remote switch,
N/A - The peer has not communicated link state for this MLAGport
Number of Multi-switch Link Aggregation Groups  : 2
Convergence control                             : Fast
```

5. Add ISC port to *VLAN*.
   Description: The ISC port must be added as a member port for any VLAN that has MLAG member ports.

   ```
   create vlan "xyz"
   ```

   On the “left” BlackDiamond 8800 switch:

   ```
   configure vlan "xyz" add port 1:1, 2:1 tagged
   ```

   On the “right” BlackDiamond 8800 switch:

   ```
   configure vlan "xyz" add port 1:2, 2:1 tagged
   ```

   Figure 18 on page 301, shows a basic MLAG network. Figure 19 shows a network with back-to-back aggregation. There is one MLAG configured on the BlackDiamond switches and three configured on the ExtremeSwitching switches.

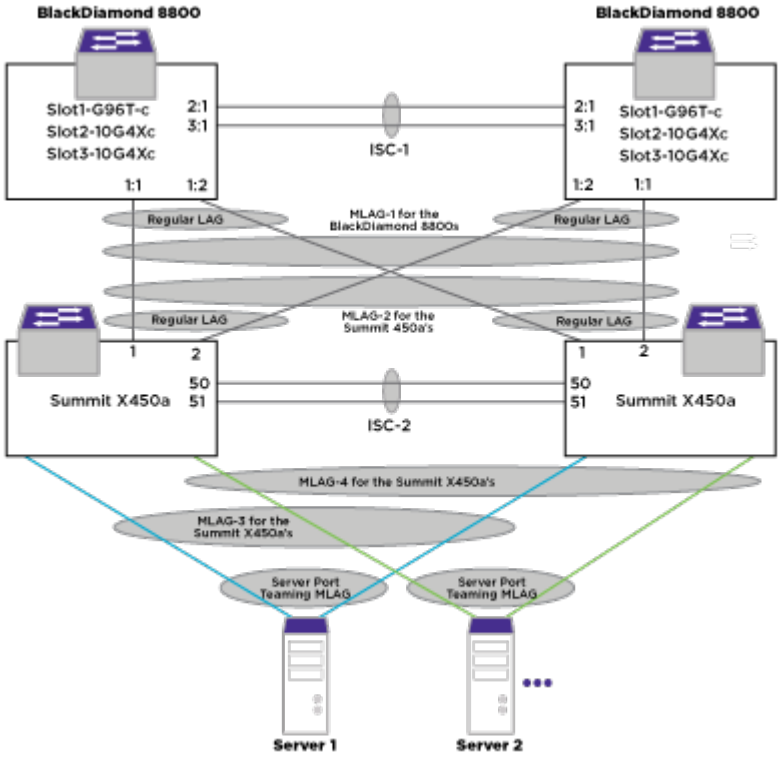


Figure 19: Two-tier MLAG Network

MLAG-LACP

Beginning in ExtremeXOS 15.3, the ExtremeXOS feature supports Link Aggregation Control Protocol (LACP) over MLAG ports. To do this, all MLAG peer switches use a common MAC in the System Identifier portion of the LACPDU transmitted over the MLAG ports. The following options and requirements are provided:

- The MLAG peer that has the highest IP address for the ISC control *VLAN* is considered the MLAG LACP master. The switch MAC of the MLAG LACP master is used as the System Identifier by all the MLAG peer switches in the LACPDUs transmitted over the MLAG ports. This is the default option.

- You can configure a common unicast MAC address for use on all the MLAG peer switches. This MAC address is used as the System Identifier by all the MLAG peer switches in the LACPDUs transmitted over the MLAG ports. This configuration is not checkpointed to the MLAG peers, and you must make sure that the same MAC address is configured on all the MLAG switches. Additionally, you must ensure that this address does not conflict with the switch MAC of the server node that teams with the MLAG peer switches.

Note

When LACP shared ports are configured as MLAG ports, a *LAG* ID change after MLAG peer reboot may result in MLAG ports being removed and re-added to the aggregator. To avoid the MLAG port flap, it is recommended to configure a common LACP MAC in both the MLAG peers using the `configure mlag peer peer_name lacp-mac lacp_mac_address` command.

LACPDU Transmission on MLAG Ports

To prevent the server node from forming two separate aggregators to the peers (which could result in a loop), it is necessary that both MLAG peers transmit LACP PDUs with the same System Identifier and Actor Key. The following points discuss how the System Identifier is determined:

- The MLAG peers must communicate at least once with each other to generate LACP PDUs on MLAG ports. If the MLAG peers do not communicate with each other, no LACP PDUs are sent out on the MLAG ports. The MLAG peers checkpoint their system MAC and checkpoint the configured MAC to the peer to determine the LACP Operational MAC.
- If no LACP MAC is configured on the MLAG peers, the LACP Operational MAC is the MAC address of the MLAG peer that has the highest IP address for ISC control *VLAN*.
- If a different LACP MAC address is configured on the MLAG peers, the configured MAC is not used. In this case, the LACP Operational MAC address is the MAC address of the MLAG peer that has the highest IP address for ISC control VLAN.
- The configured MAC address is only used when the same MAC is configured on both the MLAG peers.

Scalability Impact on Load Shared Groups

When static load sharing is used for the ports, and if there is a single link connecting the server node and the MLAG peer switches, the port does not need to be configured as a load shared port on the MLAG peer switches. Configuring LACP on MLAG ports can reduce the number of load shared ports that can be configured in the system.

Configuration Guidelines

- LACP configuration for MLAG ports (system priority, LACP timeout, activity mode, etc.) should be the same on all the MLAG peer switches.
- We recommend that the server node has a lower System Aggregation Priority than the MLAG peers so that the server node chooses which of the ports can be aggregated. As an example, there are a maximum of 8 ports that can be aggregated together, and there are eight links from Peer1 to the Server, and another eight links from Peer2 to the server node. When the server node has a lower

System Aggregation Priority, it can choose which of the total available links can be aggregated together.

- If the Port Aggregation Priority is not configured for the load shared member ports, there is a chance that only the links from server node to one of the MLAG peer are aggregated together (based on the port numbers). In this instance, the links from the server node to the other MLAG peer are unused. To avoid this, you can configure the Port Aggregation Priority on the server node so that the number of active links to the MLAG peers is balanced.
- You must configure load sharing groups on all the MLAG ports even if they contain just one port.
- It is recommended to configure static LACP MAC on both MLAG peers to avoid traffic interruption during failure scenarios.

Below are sample configurations.

Configuration on Peer1

```
create vlan "isc"
configure vlan isc tag 4000
enable sharing 5 grouping 5,10 lacp
configure vlan "isc" add ports 5 tagged
configure vlan "isc" ipaddress 1.1.1.1/8

create mlag peer "peer2"
configure mlag peer "peer2" ipaddress 1.1.1.2
configure mlag peer "peer2" lacp-mac 00:11:22:33:44:55

enable sharing 6 grouping 6,12 lacp
enable sharing 18 grouping 18 lacp
enable mlag port 6 peer "peer2" id 1
enable mlag port 18 peer "peer2" id 2
```

Configuration on Peer2

```
create vlan "isc"
configure vlan isc tag 4000
enable sharing 5 grouping 5,10 lacp
configure vlan "isc" add ports 5 tagged
configure vlan "isc" ipaddress 1.1.1.2/8

create mlag peer "peer1"
configure mlag peer "peer1" ipaddress 1.1.1.1
configure mlag peer "peer1" lacp-mac 00:11:22:33:44:55
enable sharing 20 grouping 20 lacp
enable sharing 6 grouping 6,15 lacp
enable mlag port 20 peer "peer1" id 1
enable mlag port 6 peer "peer1" id 2
```

Configuration on Server Nodes (assumed to be Extreme Switches)

```
enable sharing 1 grouping 1,2,3 lacp
configure sharing 1 lacp system-priority 100
configure lacp member-port 1 priority 10
configure lacp member-port 2 priority 20
configure lacp member-port 3 priority 15
```

Here is an illustration of the configuration:

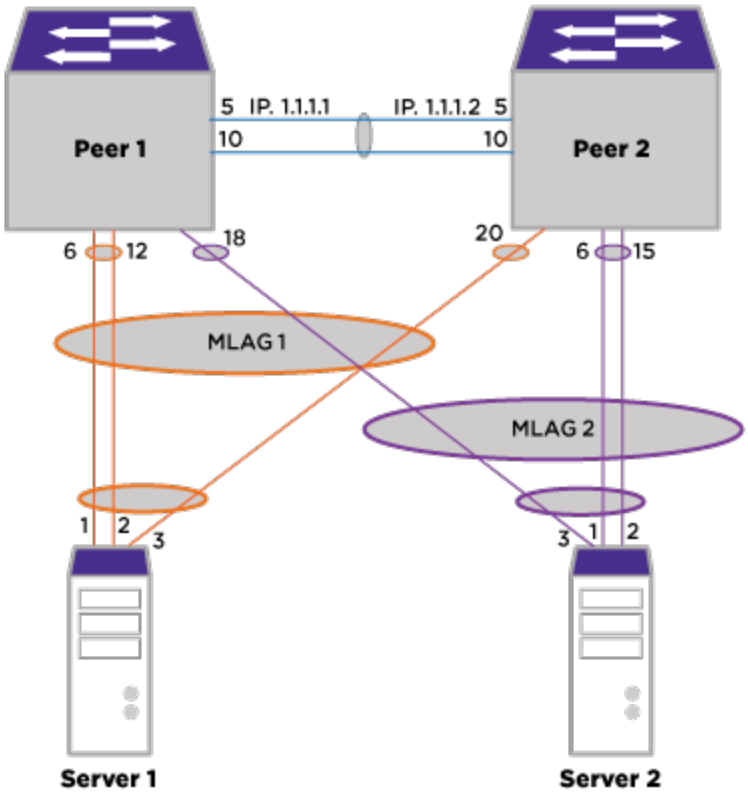


Figure 20: LACP Ports in MLAG Configuration

Mirroring

Mirroring is a function on existing Extreme Networks switches that allows copies of packets to be replicated to additional ports without affecting the normal switching functionality. The mirrored data actually occupies fabric bandwidth, so it is very likely that normal forwarding and mirroring cannot both be done at line rate. In the most general terms, traffic ingressing and/or egressing an interface is mirrored. For ports, traffic ingressing and/or egressing a port can be mirrored (refer to the `configure mirror add` command). For *VLAN*s and virtual ports, only traffic ingressing these interfaces are mirroring.

One of the common uses of the mirroring functionality is packet capture; for example sending copies of all packets that arrive on port P, vlan V, to a monitor port Q. Previous implementations of mirroring were limited to a single instance, where only one destination port (or port list) was allowed to be configured in the system. That implementation was also limited to 128 total sources of this traffic (also referred to as filters). Only VLAN and VLAN/port “filters” are currently implemented as filters.

ExtremeXOS supports Multi Instanced Mirroring that expands the number of destinations allowed to match the hardware capabilities. (Current hardware allows for up to 4 ingress mirroring instances and two egress mirroring instances.) A mirroring

instance consists of a unique destination port, or port list, and the source filters associated with it. Our current implementation allows for 128 per instance.

Note
You can have a maximum of 16 mirroring instances in the switch (including the default mirroring instance) but only 4 can be active at a time as explained below:

- Four (4) ingress
- Three (3) ingress and one (1) egress
- Two (2) ingress and two (2) egress

The maximum possible combinations for mirroring instances include:

- 2 (ingress + egress)
- 1 (ingress + egress) + 2 ingress
- 1 (ingress + egress) + 1 egress + 1 ingress

In general, there are four hardware resource slots. Each single instance uses one slot, while one (ingress + egress) instance uses two slots. So, you can use of total four slots, but there can be no more then two egress instances.

Note
You can accomplish port mirroring using ACLs, or CLEAR-Flow. See ACLs on page 341 and CLEAR-Flow for more information.

A virtual port is a combination of a VLAN and a port. The monitor port or ports can then be connected to a network analyzer for packet analysis. The system uses a traffic filter that copies a group of traffic to the monitor port(s). You can have only one monitor port or port list on the switch. This feature allows you to mirror multiple ports or VLANs to a monitor port, while preserving the ability of a single protocol analyzer to track and differentiate traffic within a broadcast domain (VLAN) and across broadcast domains (for example, across VLANs when routing).

Note
The mirroring filter limits discussed later do not apply when you are using ACLs or CLEAR-Flow.

Up to 128 mirroring filters can be configured across all active mirroring instances.

ExtremeXOS can also enables hardware mirroring of Ethernet frames to a specified remote IPv4 address, which can reside zero or more router hops away. This is useful for ExtremeAnalytics, Application Telementry, or other forms of remote network analysis or monitoring.

ExtremeXOS has enhanced mirroring to support IPFIX flows to be mirrored as well. This is done by using the mirroring capabilities in ExtremeXOS along with IPFIX to provide additional information about flows to our Application Analytics appliance (previously known as Purview). As mentioned earlier, ExtremeXOS can collect statistics about flows that are recognized based on configured flow keys. However ExtremeXOS cannot inspect packets deeper than the L4 (TCP) level. Mirroring this traffic to the Application Analytics appliance allows for deep packet inspection beyond L4 when provided a

copy of the packet payload. This enhancement provides the ability to mirror the first 15 packets of any flow to a port where Application Analytics can receive copies for deep packet inspection. As with mirroring, this allows you to configure multiple mirroring instances.

Tagging of Mirrored packets

The following conditions describe tagging of mirrored packets:

- Untagged ingress mirrored traffic egresses the monitor port(s) untagged. Tagged ingress mirrored traffic egresses the monitor port tagged.
- Egress mirrored traffic always egresses the monitor port tagged.
- On ExtremeSwitching series switches, all traffic ingressing the monitor port or ports is tagged only if the ingress packet is tagged. If the packet arrives at the ingress port as untagged, the packet egresses the monitor port or ports as untagged.

Guidelines for Mirroring

The traffic filter on ExtremeSwitching series switches can be defined based on one of the following criteria:

- Physical port—All data that traverses the port, regardless of *VLAN* configuration, is copied to the monitor port(s). You can specify which traffic the port mirrors:
  - Ingress—Mirrors traffic received at the port.
  - Egress—Mirrors traffic sent from the port.
  - Ingress and egress—Mirrors traffic either received at the port or sent from the port.

    If you omit the optional parameters, all traffic is forwarded; the default for port-based mirroring is ingress and egress.

    Note
    You can create an instance where the source is ingress only. When you add a source, pay attention to the monitor port.
- VLAN—All packets ingressing any port on a particular VLAN, regardless of the physical port configuration, is copied to the monitor port(s).
- Virtual port—All traffic ingressing the switch on a specific VLAN and port combination is copied to the monitor port(s).
- IPFIX—mirror the first 15 packets of any IPFIX flow to a port where Purview can receive a copy of the packet for deep packet inspection.
- ExtremeSwitching series switches support a maximum of 128 mirroring filters per instance.
- ExtremeXOS supports up to 16 monitor ports for one-to-many mirroring.
- Only traffic ingressing a VLAN can be monitored; you cannot specify ingressing or egressing traffic when mirroring VLAN traffic and a virtual port filter.
- In normal mirroring, a monitor port cannot be added to a load share group. In one-to-many mirroring, a monitor port list can be added to a load share group, but a loopback port cannot be used in a load share group.

- Two packets are mirrored when a packet encounters both an ingress and egress mirroring filter.
- The configuration of remote-tag does not require the creation of a VLAN with the same tag; on these platforms the existence of a VLAN with the same tag as a configured remote-tag is prevented. This combination is allowed so that an intermediate remote mirroring switch can configure remote mirroring using the same remote mirroring tag as other source switches in the network. Make sure that VLANs meant to carry normal user traffic are not configured with a tag used for remote mirroring.

  When a VLAN is created with remote-tag, that tag is locked and a normal VLAN cannot have that tag. The tag is unique across the switch. Similarly if you try to create a remote-tag VLAN where remote-tag already exists in a normal VLAN as a VLAN tag, you cannot use that tag and the VLAN creation fails.

Mirroring Rules and Restrictions

This section summarizes the rules and restrictions for configuring mirroring:

- Each configured mirror instance that you configure is saved, regardless of its state.
- To change monitor ports you must first remove all the filters.
- You cannot mirror the monitor port.
- The mirroring configuration is removed only if the configuration matches the removed *VLAN* or slot. If you have a match you can do the following:
  - Delete a VLAN (for all VLAN-based filters).
  - Delete a port from a VLAN (for all VLAN-, port-based filters).
  - Unconfigure a slot (for all port-based filters on that slot).
- Any mirrored port can also be enabled for load sharing (or link aggregation); however, each individual port of the load-sharing group must be explicitly configured for mirroring.
- When traffic is modified by hardware on egress, egress-mirrored packets may not be transmitted out of the monitor port exactly as they egressed the port containing the egress mirroring filter. Here are some examples:
  - An egress-mirrored packet that undergoes VLAN translation is mirrored with the untranslated VLAN ID.
  - Egress-mirrored IP multicast packets contain the source MAC address and VLAN ID of the unmodified packet.
  - Tagged egress VMAN packets are mirrored with the primary Ethertype as 0x8100 instead of the actual 0x88a8.
- The monitor port is automatically removed from all VLANs; you cannot add it to a VLAN.
- You cannot use the management port in mirroring configurations.
- You cannot run ELSM and mirroring on the same port. If you attempt to enable mirroring on a port that is already enabled for ELSM, the switch returns a message similar to the following: `Error: Port mirroring cannot be enabled on an ELSM enabled port.`

- With one-to-many mirroring, you need to enable jumbo frame support in the mirror-to port and loopback port, if you need to mirror tagged packets of length 1519 to 1522.
- The loopback port is dedicated for mirroring, and cannot be used for other configurations. This is indicated through the glowing LED.
- Egress mirrored packets are always tagged when egressing the monitor port. If an egress mirrored packet is untagged on the egress mirrored port, the mirrored copy contains a tag with an internal VLAN ID.
- As traffic approaches line rate, mirroring rate may decrease. Since mirroring makes copies of traffic, the bandwidth available will be devoted mostly to regular traffic instead of mirrored traffic when the load is high.
- A mirror port cannot be a *LAG*.

Configuring Mirroring

- To create a named mirror instance, use the following command:

  `create mirror` *`mirror_name`* {**`to`** [**`port`** *`port`* | **`port-list`** *`port_list`* **`loopback-port`** *`port`*] **`{ remote-tag`** *`rtag`* } | {{ } {} { [|]}{ [|]} **`priority`** *`priority_value`*]} {**`description`** *`mirror-desc`*}

- If you want to configure mirroring on multiple ports, and did not configure this with the above `create` command, use the following command using the port-list option:

  `configure mirror` *`mirror_name`* {**`to`** [**`port`** *`port`* | **`port-list`** *`port_list`* | **`loopback port`** *`port`*] {**`add`**} {{} } { [|]} { [|]}] {**`remote-tag`** *`rtag`* | **`port none`**} {**`priority`** *`priority_value`*}

  The port-list is a list of monitor ports that transmit identical copies of mirrored packets. The loopback-port is an unused port that is required when you mirror to a port-list. The loopback-port is not available for switching user data traffic. Certain L2 control protocol packets, such as *STP*, LACP, and *LLDP*, cannot be mirrored with one-to-many mirroring.

  For high availability, you can add up to four redundant remote IP addresses. If you are adding another (redundant) remote IP address to an existing mirror that already has a remote IP address configured, you must use the add option.

- Mirroring is disabled by default. To enable mirroring, use one of the following commands:
  - `enable mirror` *`mirror_name`*
  - `enable mirror to` [**`port`** *`port`* | **`port-list`** *`port_list`* **`loopback-port`** *`port`*] {**`remote-tag`** *`tag`*}
  - `enable mirror` {*`mirror_name`*} `to` **`remote-ip`** *`remote_ip_address`* {{**`vr`**} *`vr_name`*} {**`priority`** *`priority_value`*} {**`from`** [*`source_ip_address`* | **`auto-source-ip`**]} {**`ping-check`** [**`on`** | **`off`**]}]

- To disable mirroring, use the following command:
  `disable mirror [mirror_name | all]`
- To remove one or all remote IP addresses, use the following command:
  `configure mirror {mirror_name to remote-ip delete [all | remote_ip_address {{vr} vr_name}] }`

  To delete all or the last remaining remote IP address, you must disable the mirror first.

To create policy-based mirrors, see Table 103 on page 1373.

Verifying the Mirroring Configuration

The `show mirroring` command lists the ports that are involved in mirroring and identifies the monitor port. The display differs slightly depending on the platform.

Remote Mirroring

Remote mirroring enables the user to mirror traffic to remotely connected switches. Remote mirroring allows a network administrator to mirror traffic from several different remote switches to a port at a centralized location. There are two kinds of remote mirroring: Layer-2 using a remote VLAN tag, and Layer-3 using GRE encapsulation through a routed network to a remote IP address. This section describes remote mirroring using Layer-2. Remote mirroring is accomplished by reserving a dedicated *VLAN* throughout the network for carrying the mirrored traffic. You can enable remote mirroring on the ExtremeSwitching series switches.

Figure 21 shows a typical remote mirroring topology. Switch A is the source switch that contains ports, VLANs, and/or virtual ports to be remotely mirrored. Port 25 is the local monitor port on Switch A. Switch B is the intermediate switch. Switch C is the destination switch, which is connected to the network analyzer.

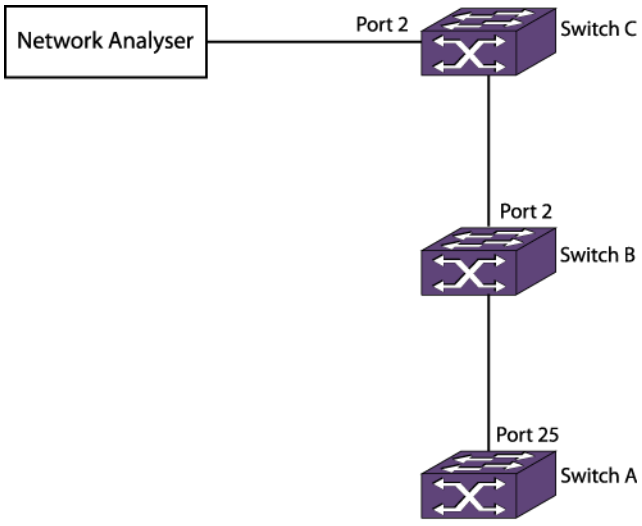


Figure 21: Remote Mirroring Topology

All the mirrored packets are tagged with the **remote-tag** specified by the source switch, whether the packet is already tagged or not. The intermediate switches forward the remote-tagged mirrored packets to the adjacent intermediate/destination switch, as these ports are added as tagged. The port connected to the network analyzer is added as untagged in the destination switch. This causes the destination switch to remove the remote-tag, and the mirrored packet reaches the network analyzer as the source switch sent it.

Unlike basic mirroring, remote mirroring does not remove VLAN membership from the local monitor port(s). This allows remote mirroring to use the existing network topology to transport remote mirrored packets to a destination switch.

Configuration Details

This section describes in detail the configuration details for the topology shown in Figure 21 on page 312.

Configuring the Source Switch

The remote-tag keyword followed by the tag is added in the command to enable mirroring. For example, you can establish ports 24 and 25 as monitor ports, from which any mirrored packets are transmitted with an additional *VLAN* tag containing a VLAN ID of 1000:

Remote mirroring can also be enabled to a single port, without the port-list and loopback-port keywords.

- To establish ports 24 and 25 as monitor ports, follow the example:

  ```
  enable mirror to port-list 24,25 loopback-port 1 remote-tag 1000
  ```

  The `show mirror` output displays the remote tag when remote mirroring is configured.
- To enable remote mirroring to port 25, follow the example:

  ```
  enable mirror to port 25 remote-tag 1000
  ```

Configuring the Intermediate Switch

Reserve a *VLAN* with the remote-mirroring keyword in all the intermediate switches for remote mirroring. When you enable mirroring with remote-tag 1000, you need to reserve a VLAN with tag 1000 in all the intermediate switches for remote mirroring. The remote mirroring VLAN in the intermediate switches is used for carrying the mirroring traffic to the destination switch. The ports connecting the source and destination switches are added as tagged in the intermediate switches.

Another way to configure a remote mirroring VLAN is to create a normal VLAN and disable learning on the VLAN. *IGMP* snooping must be disabled on that VLAN for you to remotely mirror multicast packets through the switch.

- You may add the remote-mirroring keyword when you configure the tag to differentiate a normal VLAN from the remote mirroring VLAN.

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000 remote-mirroring
configure vlan remote_vlan add ports 1,2 tagged
```

  Using the remote-mirroring keyword automatically disables learning and IGMP snooping on the VLAN.

- You may use the following configuration for creating the remote mirroring VLAN:

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000
disable learning vlan remote_vlan
disable igmp snooping remote_vlan
```

Configuring the Destination Switch

Remote mirroring *VLAN* on the destination switch. The configuration on the destination switch is same as that of the intermediate switches, except that the port connected to the network analyzer is added as untagged whereas all the other ports connected to the switches are added as tagged.

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000 remote-mirroring
configure vlan remote_vlan add ports 1 tagged
configure vlan remote_vlan add ports 2 untagged
```

For a remote mirroring VLAN, the configured tag displayed by the show vlan output is remote tag instead of the normal tag.

Remote Mirroring Guidelines

The following are guidelines for remote mirroring:

- Configurations of remote mirroring, which might cause protocol packets to be remotely mirrored, are not recommended. Since all packet types are mirrored when you configure remote mirroring, remotely mirrored protocol packets may have undesirable affects on intermediate and destination switches. Blocking *EDP* packets on a remote mirroring *VLAN* is one example of a case where you must perform an extra action to accommodate the remote mirroring of protocol packets.

  For EDP configuration on the remote mirroring VLAN, in the intermediate and destination switches you need to install *ACL* to block the EDP packets on the remote mirroring VLAN. Use the following commands for installation:

```
create access-list remote_edp " ethernet-destination-address 00:e0:2b:00:00:00 mask
ff:ff:ff:ff:ff:ff ;" "deny"
conf access-list add "remote_edp" first vlan "remote_vlan"
```

Using Remote Mirroring with Redundancy Protocols

You can use remote mirroring with one-to-many mirroring to provide a redundant path from the source switch to the destination switch. Using *EAPS* or Spanning Tree can provide remote mirroring packets a redundant loop-free path through the network. You should perform the configuration of EAPS or Spanning Tree before adding mirroring filters on the source switch to prevent looping.

Remote Mirroring with EAPS

In Figure 22, the traffic from switch A is mirrored to the two ports 8:2 and 1:48 to connect to the destination switch. Using the configuration shown, remote mirrored packets have a loop-free redundant path through the network using *EAPS*.

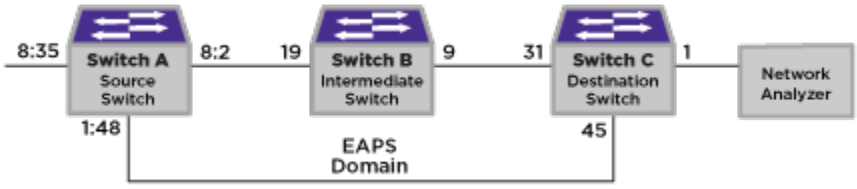


Figure 22: Remote Mirroring with EAPS

The configuration for the topology in the above figure is given in the following sections.

Switch A Configuration

```
enable mirror to port-list 8:2,1:48 loopback-port 8:1 remote-tag 1000
configure mirror add port 8:35 create vlan eaps_control
configure vlan eaps_control tag 1001
configure vlan eaps_control add ports 8:2,1:48 tag
create eaps eaps1
configure eaps1 mode master
configure eaps1 primary port 8:2
configure eaps1 secondary port 1:48
configure eaps1 add control eaps_control
configure eaps1 add protected internalMirrorLoopback
enable eaps1
enable eaps
```

Switch B Configuration

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000 remote-mirroring
configure vlan remote_vlan add ports 19,9 tag
create vlan eaps_control
configure vlan eaps_control tag 1001
configure vlan eaps_control add ports 19,9 tag
create eaps eaps1
configure eaps1 mode transit
configure eaps1 primary port 19
configure eaps1 secondary port 9
configure eaps1 add control eaps_control
configure eaps1 add protected remote_vlan
enable eaps1
enable eaps
```

Switch C configuration

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000 remote-mirroring
configure vlan remote_vlan add ports 31,45 tag
configure vlan remote_vlan add ports 1
create vlan eaps_control
configure vlan eaps_control tag 1001
configure vlan eaps_control add ports 31,45 tag
create eaps eaps1
configure eaps1 mode transit
configure eaps1 primary port 31
configure eaps1 secondary port 45
configure eaps1 add control eaps_control
configure eaps1 add protected remote_vlan
enable eaps1
enable eaps
```

Note

The internalMirrorLoopback is an internal VMAN created when enabling mirroring to multiple ports. Depending on the platform, the internal *VLAN* or VMAN needs to be added as the protected VLAN in the source switch in order to block the ports for mirroring when EAPS is complete.

Remote Mirroring With STP

For the same topology shown in Figure 22 on page 315, you can use *STP* instead of using *EAPS*. A sample configuration follows.

Switch A Configuration

```
enable mirror to port-list 8:2,1:48 loopback-port 8:1 remote-tag 1000
configure mirror add port 8:35
create vlan v1
configure vlan v1 tag 1001
configure vlan v1 add ports 8:2,1:48 tag
create stp stp1
configure stp1 mode dot1w
configure stp1 add v1 ports all
configure stp1 tag 1001
configure stp1 add vlan internalMirrorLoopback ports 8:2,1:48
enable stp1
enable stpd
```

Switch B Configuration

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000 remote-mirroring
configure vlan remote_vlan add ports 19,9 tag
create vlan v1
configure vlan v1 tag 1001
configure vlan v1 add ports 19,9 tag
create stp stp1
configure stp1 mode dot1w
configure stp1 add v1 ports all
configure stp1 tag 1001
configure stp1 add vlan remote_vlan ports all
enable stp1
enable stpd
```

Switch C Configuration

```
create vlan remote_vlan
configure vlan remote_vlan tag 1000 remote-mirroring
configure vlan remote_vlan add ports 31,45 tag
configure vlan remote_vlan add ports 1
create vlan v1
configure vlan v1 tag 1001
configure vlan v1 add ports 31,45 tag
create stp stp1
configure stp1 mode dot1w
configure stp1 add v1 ports all
configure stp1 tag 1001
configure stp1 add vlan remote_vlan ports 31,45
enable stp1
enable stpd
```

Remote Mirroring Using Layer 3

To use remote mirroring using Layer 3, enable IPv4 forwarding along the routed path to the device collecting GRE-encapsulated mirrored packets, also known as ERSPAN. Then, configure the collector device's IP address as the remote IP address in the following command:

enable mirror {*mirror_name*} to **remote-ip** *remote_ip_address* {{**vr**} *vr_name*} {**priority** *priority_value*} {**from** [*source_ip_address* | **auto-source-ip**]} {**ping-check** [**on** | **off**]}]

To configure the GRE protocol type, use the command:

configure **mirror to remote-ip protocol-type** [**erspan-v1** | **trans-ether-bridging** | **user-defined** *protocol_value*]

Redundant Remote IP Address Mirroring

For high availability, each instance of a mirror to a remote IP address can have up to four remote IP addresses configured with a unique priority—from 1 (least preferred) to 100 (most preferred), with a default of 50. For each mirror instance, the remote IP address with the highest priority value that is deemed “up” is used as the destination IP address for GRE-tunneled mirrored traffic. All other remote IP addresses deemed “up” for that mirror instance are on standby, ready to be used in the event the preferred remote IP address goes down. Each remote IP address is monitored independently with ping health check, route tracking, and gateway ARP tracking.

To add more redundant remote IP addresses, use the following command using the remote-ip , add, and priority options:

configure mirror *mirror_name* {**to** [**port** *port* | **port-list** *port_list* | **loopback port** *port*] {**add**} {{} } { [|]} { [|]}] {**remote-tag** *rtag* | **port none**} {**priority** *priority_value*}

Extreme Discovery Protocol

The *EDP* is used to gather information about neighbor Extreme Networks switches. EDP is used by the switches to exchange topology information. Information communicated using EDP includes:

- Switch MAC address (switch ID)
- Switch software version information
- Switch IP address
- Switch *VLAN* IP information
- Switch port number
- Switch configuration data: duplex and speed
- Local and neighbor VLAN IDs

EDP is enabled on all ports by default, including the management port. EDP enabled ports advertise information about the Extreme Networks switch to other switches on the interface and receives advertisements from other Extreme Networks switches. Information about other Extreme Networks switches is discarded after a timeout interval is reached without receiving another advertisement.

Starting with ExtremeXOS 30.3, EDP detects mismatches when untagged ports are used in configurations where the port endpoints are used in VLANs that have different VLAN IDs. While this is not technically invalid, it nearly always indicates a mis-configuration. If a mismatch ID is detected, a log event is generated. The event is only generated once per boot. There is also a command to display port/VLAN ID endpoints for easy comparison (`show edp {ports [all | ports] {detail | vlan-id {mismatch {untagged}} {neighbor nbr}}}`).

- To disable EDP on specified ports, use the following command:

  `disable edp ports [ports | all]`
- To enable EDP on specified ports, use the following command:

  `enable edp ports [ports | all]`
- To clear EDP counters on the switch, use the following command:

  `clear counters edp`

  This command clears the following counters for EDP protocol data units (PDUs) sent and received per EDP port:
  - Switch PDUs transmitted
  - VLAN PDUs transmitted
  - Transmit PDUs with errors
  - Switch PDUs received
  - VLAN PDUs received
  - Received PDUs with errors

- To view EDP port information on the switch, use the following command:

  `show edp {ports [all | ports] {detail | vlan-id {mismatch {untagged}} {neighbor nbr}}}`

- To configure the advertisement interval and the timeout interval, use the following command:

  `configure edp advertisment-interval timer holddown-interval timeout`

For information about displaying EDP status, see Displaying Port Information on page 331.

ExtremeXOS Cisco Discovery Protocol

Network elements exchange various information with neighboring elements that ranges from the switch IP/Mac addresses, switch names, versions of software, etc. This information is maintained in an Information Base, and management applications use it to create a topology map and manage the entire network. Vendor-specific discovery protocols are used to learn about network neighbor elements and their capabilities, and to exchange this information. Cisco Discovery Protocol (CDP) is a Cisco® proprietary protocol that runs between direct connected network entities (routers, switches, remote access devices, IP telephones) to provide information sharing capabilities. ExtremeXOS supports CDP, and is called EXOS-CDP.

Limitations

The ExtremeXOS support for CDP has the following limitations:

- *SNMP* for this feature is not supported.
- You should use this feature mainly for the network-endpoint devices, but you can also use it in network-network devices that have CDP support.
- When port access is controlled by net login, the port must be authorized before it receives CDP packets.

Platform Support

This feature is supported on all ExtremeXOS platforms.

EXOS CDP Support

The ExtremeXOS-CDP implementation gathers information about network neighbors that support the Cisco Discovery protocol. This includes edge devices (VoIP phones) in the network-edge domain.

EXOS-CDP runs on top of the controlled port of an 802 MAC client. If port access is controlled by IEEE 802.1X, the port must be authorized before you enable CDP protocol receive functionality. CDP also runs over an aggregated MAC client, and the CDP protocol information must run over all the physical MAC clients of the aggregated ports. The spanning tree state of a port does not affect the transmission of CDP PDUs.

Each CDP message contains information identifying the source port as a connection endpoint identifier. It also contains at least one network address that can be used by an NMS to reach a management agent on the device (through the indicated source port). Each CDP message contains a time-to-live value, which tells the recipient when to discard each element of learned topology information.

By default EXOS-CDP feature is disabled.

Both CDP version 1 and version 2 are supported in EXOS-CDP implementation.

CDP Packet Format

The CDP control packets are encapsulated to the Sub-network Access Protocol (SNAP), and are sent as multicasts with Cisco®-defined multicast MAC address 01:00:0C:CC:CC:CC.

The following table lists and describes the CDP packet fields.

Field	Description
Version	Version of the CDP being used values 0x1 or 0x02
Time-to-Live (TTL)	TTL indicates the amount of time in seconds that a receiver should retain the information contained in this packet.
Check Sum	Standard IP checksum
Type	Indicates the type of the TLV
Length	Indicates the total length of the Type,Length,Value fields
Value	Indicates the value of the TLV

Different Types of TLVs

The following list identifies the different types of TLVs.

- Device ID Information TLVs

  This TLV is used to identify Device name in form of Character string. In EXOS-CDP implementation it will be the configurable value. System Mac address will be the default Device ID. Both receive and transmit of this TLV is supported.

- Address Information TLVs

  This TLV contains a number that indicates how many addresses are contained in the packet, followed by one entry for each address being advertised. The addresses advertised are the ones assigned to the interface on which the CDP message is sent. A device can advertise all addresses for a given protocol suite and, optionally, can advertise one or more loopback IP addresses. If the device can be managed by *SNMP*, the first entry in the address type/length/value is an address at which the device receives SNMP messages. Maximum of 32 IP address are supported. Both receive and transmit of this TLV is supported.

The following table identifies and describes the various fields in the Address Information TLV.

Field	Description
Protocol Type	Protocol type It can be one of the following values 1. NLPID 2. 802.2 format
Length	Length of the protocol field.
Protocol	One of the following values: ◦ 0x81—ISO CLNS (protocol type 3D 1) ◦ 0xCC—IP (protocol type 3D 1) ◦ 0xAAAA03 000000 0800—Pv6 (protocol type 3D 2) ◦ 0xAAAA03 000000 6003—DECNET Phase IV (protocol type 3D 2) ◦ 0xAAAA03 000000 809B—AppleTalk (protocol type 3D 2) ◦ 0xAAAA03 000000 8137—Novell IPX (protocol type 3D 2) ◦ 0xAAAA03 000000 80c4—Banyan VINES (protocol type 3D 2) ◦ 0xAAAA03 000000 0600— XNS (protocol type 3D 2) ◦ 0xAAAA03 000000 8019—Apollo Domain (protocol type 3D 2)
Address Length	Length of the address fields in bytes
Address	Address of the interface or the address of the system if addresses are not assigned to the interface.

- Port ID TLVs

  The port ID contains an ASCII character string that identifies the port on which the CDP message is sent. Both receive and transmit of this TLV is supported.

- Capabilities TLVs

  The capability TLV describes the device's functional capability. It can be set to one of the bits listed below. Both receive and transmit of this TLV is supported.

  The following table identifies and describes the various bits in the Capabilities TLV.

Bit Description	Bit Description
0x01	Performs level 3 routing for at least one network layer protocol.
0x02	Performs level 2 transparent bridging.
0x04	Performs level 2 source-route bridging. A source-route bridge would set both this bit and bit 0x02.
0x08	Performs level 2 switching. The difference between this bit and bit 0x02 is that a switch does not run the *STP*.
0x10	Sends and receives packets for at least one network layer protocol. If the device is routing the protocol, this Bit should not be set.

Bit Description	Bit Description
0x20	The bridge or switch does not forward *IGMP* Report packets on non-router ports.
0x40	Provides level 1 functionality.

- Version TLV

  The version TLV contains a character string that provides information about the software release version that the device is running. In EXOS-CDP version will be software version. Both receive and transmit of this TLV is supported.

- Platform TLV

  The platform TLV contains an ASCII character string that describes the hardware platform of the device. These platform TLV values are EXOS platform information that is the same seen when you issue the `show switch` command. Both receive and transmit of this TLV is supported.

- Native *VLAN* TLV

  The native VLAN TLV indicates, per interface, the assumed VLAN for untagged packets on the interface. CDP learns the native VLAN for an interface. Both receive and transmit of this TLV is supported.

- Duplex TLV

  The duplex TLV indicates the duplex configuration of the interface. Both receive and transmit of this TLV is supported.

- Voice VLAN Query TLV

  The voice VLAN query TLV indicates what VLAN ID is to be used for voice. This TLV is mostly used in IP Phones to configure the voice VLAN. IP Phone will query the Neighbor switch for voice VLAN. The switch will reply the VLAN ID value by transmitting in CDP PDU with type voice vlan reply TLV. Only receive functionality is supported.

- Voice VLAN Reply TLV

  The voice VLAN reply TLV indicates the voice VID for the VOIP VLAN Query. The switch will reply the VLAN ID value when it receives the voice vlan query TLV. Only transmit functionality is supported. When the VOIP VLAN Query is received immediately a CDP PDU is sent with reply TLV filled.

- Power TLV

  The power TLV indicates the amount of power consumed by remote devices. Only receive functionality is supported.

- Power Request TLV

  The power request TLV indicates the power requested by the powered device (PD). Only the receive functionality of this TLV is supported.

- Power Available TLV

The power available TLV indicates the power available in the powered source device. This information will be sent in the CDP PDU to the powered device (PD). Only the transmission of this TLV is supported.

- MTU TLV

  The MTU TLV indicates the size of the largest datagram that can be sent/received by a remote device. Only receive functionality is supported.

- Trust Bit TLV

  The trust bit TLV offers the ability to specify whether the PC port of the phone is trusted. Only transmit functionality is supported. This value is configured by CLI command to say whether the port is trusted/untrusted. Only the transmission of this TLV is supported.

- Untrust QOS TLV

  The untrust QOS TLV offers the ability to specify whether the PC port of the phone is untrusted then what to mark the L2 packet cos bit. Only the transmission of this TLV is supported. This value is configured by user through CLI command.

- System Name TLV

  The system name TLV indicates the value of the remote device's sysName MIB object. Both transmit and receive functionality is supported.

- System Object ID TLV

  The system object ID TLV indicates the value of the remote device's sysObjectID MIB. Only receive functionality is supported.

- Management Address TLV

  The management address indicates the IP address for which the device accepts the SNMP messages. Both transmit and receive functionality is supported. CDP advertises Management Address TLV with only one IP address.

  To configure a specified VLAN's IP address as the management address to be advertised by CDP, use the following command:

  ```
  configure cdp management-address [{vlan} vlan_name | vlan vlan_id]
  {primary-ip | secondary-ip secondary_ip_address}]
  ```

- Location TLV

  The location TLV indicates the physical location, which says either the postal pin address. Both transmit and receive functionality is supported. The location value will be configured by `configure snmp syslocation sysLocation`

EXOS CDP Protocol Operations

The ExtremeXOS CDP implementation performs the following tasks:

- Adds a filter for CDP Multicast MAC when the CDP protocol is enabled on port.
- Processes received CDP messages.

- Transmits CDP messages to peer when CDP is enabled.
- Maintains the current neighbor's information in the local database.

Protocol Operations

- Multicast Address

  This EXOS-CDP Implementation describes the operation of the protocol for reception of CDPPDUs on a single port, using a Multicast Mac address as the destination (01:00:0C:CC:CC:CC).
- CDP Initialization

  The following EXOS CDP related functions are created as part of the existing CDP process.
  - Upon system initialization, the appropriate default values are assigned to CDP global, port, and neighbor data structure.
  - Retrieval of non-volatile configuration values.
- CDP Frame Reception

  Frames that are recognized as CDP frames are validated to determine whether they are properly constructed and contain the correct CDP packet format. Frames that pass validation criteria are used to update the contents of the neighbor entries in the database.

  If the neighbor entry does not exist in the CDP neighbor database, a new entry is created. If the neighbor entry already exists in the database, new information contained in the CDP PDU is used to replace the changed information of the existing entry in the neighbor database.

  One of the parameters received in an incoming CDP PDU is the TTL value. This determines how long the information is stored in the neighbor database before it is aged out and deleted. Aging out old data ensures that neighbor entries are purged that were originated by systems that are no longer neighbors, either because of system failure or system inactivation.
- CDP Port/Connection Failure

  If the port, or the connection to the remote system fails, the neighbor entity from the database is not deleted. It will be deleted after the associated TTL timer expires.
- Timers

  The timers are started when at least one neighbor is created. Once the timer expires, it checks the neighbor database and deletes the neighbor entry when the hold time reaches 0.
- CDP PDU Validation

  The following attributes are validated in the received CDP PDU:
  - The checksum must be re-calculated in order for the packet to be valid. Otherwise, the EXOS-cdp will reject the packet. Standard IP checksum mechanism is used for calculating the checksum
  - The receiving CDP packet length must be greater than, or equal to 4.

- The packet version must be 1 or 2.

- **CDP Frame Transmission**

  Under normal circumstances, remote devices send CDP messages at a configured interval (default 1 minute). If you enable CDP between CDP Messages in Extreme Switches, the feature must wait for next CDP PDU to update in the neighbor database. To avoid this delay when CDP is enabled, a CDP PDUs is sent to peer.

  For every default configuration of message interval 60 seconds, CDP packets are transmitted on the CDP-enabled ports.This message interval is the configurable through CLI. The default value is 60 seconds, and the range for this message interval is 5 to 254 seconds

- **Storing CDP Neighbors**

  Advertisements received from the peer contain time-to-live information that indicates the length of time a receiving device holds CDP information before discarding it.

  A neighbor entry is discarded after three advertisements from the device are missed.

- **CDP Packet Statistics**

  CDP counters of total TX and RX CDP control packets are maintained on each port.

- **CDP and Link Aggregation**

  CDP will run on all member ports of the link aggregated ports. Transmit and receive of CDP control packets will be for each individual port. CDP will be enabled for each individual port of the link aggregated ports.

  For example, if ports 1,2,3 are part of link aggregation, then the administrator needs to enable CDP on Port 1, Port 2, and Port 3. Rx and Tx of CDP will be enabled for ports 1, 2, and 3.

Software-Controlled Redundant Port and Smart Redundancy

Using the software-controlled redundant port feature, you can back up a specified Ethernet port (primary) with a redundant, dedicated Ethernet port. Both ports must be on the same switch.

If the primary port fails, the switch will establish a link on the redundant port and the redundant port becomes active. Only one side of the link must be configured as redundant because the redundant port link is held in standby state on both sides of the link. This feature provides very fast path or network redundancy.

Note
You cannot have any Layer 2 protocols configured on any of the VLANs that are present on the ports.

The Smart Redundancy feature allows control over how the failover from a redundant port to the primary port is managed. If this feature is enabled, which is the default setting, the switch attempts to revert to the primary port as soon as it can be recovered.

If the feature is disabled, the switch attempts only to recover the primary port to active if the redundant port fails.

A typical configuration of software-controlled redundant ports is a dual-homed implementation (shown in Figure 23). This example maintains connectivity only if the link between switch A and switch B remains open; that link is outside the scope of the software-controlled port redundancy on switch C.

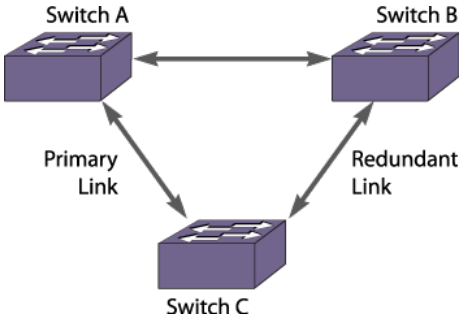


Figure 23: Dual-Homed Implementation for Switch C

In normal operation, the primary port is active and the software redundant switch (switch C) blocks the redundant port for all traffic, thereby avoiding a loop in the network. If the switch detects that the primary port is down, the switch unblocks the redundant port and allows traffic to flow through that redundant port.

Note
The primary and redundant ports must have identical *VLAN* membership.

You configure the software-controlled redundant port feature either to have the redundant link always physically up but logically blocked, or to have the link always physically down. The default value is to have the link physically down, or Off.

By default, Smart Redundancy is always enabled. If you enable Smart Redundancy, the switch automatically fails over to the redundant port and returns traffic to the primary port after connectivity is restored on that port. If you do not want the automatic restoration of the primary link when it becomes active, disable Smart Redundancy.

Guidelines for Software-Controlled Redundant Ports and Port Groups

Software-controlled redundant ports and port groups have the following guidelines and limitations:

- You cannot have any Layer 2 protocols configured on any of the *VLAN*s that are present on the ports. (You will see an error message if you attempt to configure software redundant ports on ports with VLANs running Layer 2 protocols.)
- The primary and redundant ports must have identical VLAN membership.
- The master port is the only port of a load-sharing group that can be configured as either a primary or redundant port. Also, all ports on the load-sharing group must fail before the software-controlled redundancy is triggered.
- You must disable the software redundancy on the master port before enabling or disabling load sharing.

- You can configure only one redundant port for each primary port.
- Recovery may be limited by *FDB* aging on the neighboring switch for unidirectional traffic. For bi-directional traffic, the recovery is immediate.

Configuring Software-Controlled Redundant Ports

When provisioning software-controlled redundant ports, configure only one side of the link as redundant. To enable the software-controlled redundant port feature, the primary and redundant ports must have identical *VLAN* membership.

In the following figure only the ports on switch C would be configured as redundant.

- To configure a software-controlled redundant port, use the following command:

  `configure ports` *`primaryPort`* **`redundant`** *`secondaryPort`* {**`link`** [**`on`** | **`off`**]}

  The first port specified is the primary port. The second port specified is the redundant port.
- Unconfigure a software-controlled redundant port, use the following command and enter the primary port(s):

  `unconfigure ports` *`port_list`* **`redundant`**
- To configure the switch for the Smart Redundancy feature, use the following command:

  `enable smartredundancy` *`port_list`*
- To disable the Smart Redundancy feature, use the following command:

  `disable smartredundancy` *`port_list`*

Verifying Software-Controlled Redundant Port Configurations

You can verify the software-controlled redundant port configuration by issuing a variety of commands.

- To display the redundant ports as well as which are active or members of load-sharing groups, use the following command:

  `show ports redundant`
- To display information on which ports are primary and redundant software-controlled redundancy ports, use the following command:

  `show port` {**`mgmt`** | *`port_list`* | **`tag`** *`tag`*} **`information`** {**`detail`**}

Refer to Displaying Port Information on page 331 for more information on the `show port information` command.

Port Configuration Reset to Factory Defaults

Version 33.6.1 adds the ability to reset all port-level configurations to factory defaults with a single command. This feature is useful when repurposing ports, troubleshooting configuration issues, or returning ports to a known baseline state.

What Gets Reset

The **unconfigure ports <port_list> configuration** command resets all port-related settings to factory defaults, including:

- VLAN membership (port returns to default VLAN)
- Virtual Router (VR) membership
- Load sharing group membership
- Policy configurations and assignments
- Protocol settings (STP, LLDP, etc.)
- Port speed and duplex settings
- Flow control settings
- QoS configurations
- Security features (port security, MAC locking, etc.)
- Jumbo frame settings
- All other per-port feature configurations

After executing this command, the port configuration matches the state immediately after an **unconfigure switch** command.

Resetting Port Configuration

To reset one or more ports to factory defaults, use the following command:

```
unconfigure ports <port_list> configuration
```

Where *port_list* specifies one or more ports. You can specify individual ports, port ranges, or comma-separated lists.

Examples:

```
# Reset a single port
unconfigure ports 5 configuration

# Reset a range of ports
unconfigure ports 10-20 configuration

# Reset multiple non-contiguous ports
unconfigure ports 1,5,10-15,25 configuration
```

Confirmation Prompt

The command displays a warning message and requires confirmation before proceeding:

```
Warning: This will restore factory defaults to all port-related configuration
for selected port(s). Do you wish to proceed? (y/N)
```

Enter `y` or `Yes` to proceed with the reset operation. Enter `N`, `No`, or press Enter to cancel the operation.

This confirmation prompt prevents accidental configuration loss.

Use Cases

This command is particularly useful in the following scenarios:

- **Port repurposing:** When moving a port from one function to another (for example, from a trunk port to an access port), resetting ensures no residual configuration interferes with the new purpose.
- **Troubleshooting:** When diagnosing port issues, resetting to a known baseline state eliminates configuration as a variable.
- **Device migration:** When moving devices between ports, ensuring the destination port has default configuration prevents conflicts.
- **Configuration cleanup:** After testing or temporary configurations, quickly restore ports to their default state.
- **Stack member replacement:** When replacing a failed stack member, reset ports on the new member to match the expected default state.

Impact on Traffic

Resetting port configuration disrupts traffic on the affected ports. The ports briefly go down and return to an operational state with default configuration. Plan to execute this command during a maintenance window or when traffic disruption is acceptable.

Warning
If the port is part of a critical path (such as an uplink or part of a load sharing group), resetting the configuration causes network interruption. Ensure you have documented the required configuration and are prepared to reconfigure the port immediately if necessary.

Verifying Port Configuration

After resetting port configuration, verify the port status and configuration using standard show commands:

```
# View port status
show ports <port_list> information detail

# View VLAN membership
show vlan <port_list>

# View protocol-specific configuration
show lldp ports <port_list>
show stp ports <port_list>
```

Related Commands

The **unconfigure ports** command has several other options for unconfiguring specific features:

- **unconfigure ports <port_list> description-string** - Removes the port description
- **unconfigure ports <port_list> display-string** - Removes the display string

- **unconfigure ports <port_list> link-flap-detection** - Resets link flap detection parameters

The **configuration** option differs from these by resetting *all* port-related settings, not just specific features.

Configuring Automatic Failover for Combination Ports

ExtremeSwitching Switches with Shared Copper/Fiber Gigabit Ports Only

On ExtremeSwitching switches with shared copper/fiber gigabit ports, you configure automatic failover using the combination ports. These ports are called combination ports because either the fiber port or the copper port is active, but they are never active concurrently. These ports, also called redundant ports, are shared PHY copper and fiber ports.

If you plan to use the automatic failover feature, ensure that port settings are set correctly for autonegotiation.

Note
You may experience a brief episode of the link going down and recovering during the failover.

- To display the port type currently used as well as the preferred media setting, use the following command:

  `show port {mgmt | port_list | tag tag} information {detail}`

  Refer to Displaying Port Information on page 331 for more information on the `show port information` command.

  Hardware determines when a link is lost and swaps the primary and redundant ports to maintain stability.

  After a failover occurs, the switch keeps the current port assignment until there is another failure or until you change the assignment using the CLI.
- To change the uplink failover assignment, use the following command:

  `configure ports port_list preferred-medium [copper | fiber] {force}`

  The default preferred-medium is fiber. If using the force option, it disables automatic failover. If you force the preferred-medium to fiber and the fiber link goes away, the copper link is not used, even if available.

  Note
  When the Tx of active fiber link is disconnected while Rx is still connected, the fiber link goes down but the copper link does not come up.

  For more information about combination ports on ExtremeSwitching switches, see the *hardware installation guide* for your switch.

Displaying Port Information

You can display summary port configuration information. The following commands display summary port configuration information:

```
show ports {mgmt | port_list | tag tag} configuration {no-refresh}
```

```
show port {mgmt | port_list | tag tag} information {detail}
```

The `show ports configuration` command shows either summary configuration information on all the ports, or more detailed configuration information on specific ports. If you specify the **no-refresh** parameter, the system displays a snapshot of the data at the time you issue the command.

The `show ports information` command shows you either summary information on all the ports, or more detailed information on specific ports. The output from the command differs very slightly depending on the platform you are using.

To display real-time port utilization information, use the following command:

```
show ports {mgmt | port_list | tag tag | stack-ports stacking-port_list}
utilization {bandwidth | bytes | packet}
```

To display the remote end of the link's advertised auto-negotiation ability, use the following command:

```
show ports {port_list} advertised
```

Note
This command is applicable only for port speed of up to 1Gbps This command is not applicable for Extended Edge Switching bridge port extenders (BPE) ports.

When you use a parameter (packets, bytes, or bandwidth) with the above command, the display for the specified type shows a snapshot per port when you issue the command.

Optic Port Compatibility Check

For QSFP/QSFP28 (non-stacking) ports on all supported platforms, the **show port information** and **show port information detail** commands identify whether a port is properly configured for the required partition or speed, or if there is not appropriate licensing support. For example, these commands will indicate:

- if a QSFP28 to 4xSFP28 breakout cable has the QSFP28 end inserted into a port that is misconfigured for 1x100G partition mode; or
- if a 1x100G QSFP28 optic is inserted into a port that is not licensed to run at the indicated higher speed

The "?" symbol is used to indicate an optic with a mismatched partition. If an optic is a third party unsupported optic, then the "!" unsupported indication will override the "?" mismatched partition indication.

If the switch does not have the proper license to operate at the optic's given speed, then the port cannot be partitioned for any corresponding port speeds. For example, if a 100G optic is inserted into an unlicensed port regardless of the operating mode, the CLI will display "?$" to indicate a mismatched partition and unlicensed port. In addition to these CLI display indications, a corresponding informational message will be logged to help determine the cause of the indications.

To show mismatched partitions, run the following commands:

```
show port {mgmt |port_list | tag tag} information {detail}
```

```
show ports {mgmt | port_list | tag tag} configuration {no-refresh |
refresh}
```

ExtremeXOS Port Description String

ExtremeXOS provides a configurable per-port "display-string" parameter that is displayed on each of the `show port` CLI commands, exposed through the *SNMP* ifAlias element, and accessible through the XML port.xsd API. Previously, the field was limited to 20 characters. This feature provides a new and separate per-port field call "description-string" that allows you to configure strings up to 255 characters.

Some characters are not permitted as they have special meanings. These characters include the following:

- "
- <
- >
- :
- <space>
- &.

The first character should by alphanumeric.

This new field is CLI accessible only through the `show port info detail` command, but you can also access it through the SNMP ifAlias object of IfXTable from IF-MIB (RFC 2233), and through the XML API.

Configure Port decription-string

To configure up to 255 characters associated with a port, issue the command:

```
configure port port_list description-string string
```

If you configure a string longer than 64 characters, the following warning will be displayed: `Note: Port description strings longer than 64 chars are only accessible via SNMP if the following command is issued: configure snmp ifmib ifalias size extended`

To control the accessible string size (default 64, per MIB) for the *SNMP* ifAlias object, issue the command:

```
config snmp ifmib ifalias size [default | extended]
```

If you choose the extended size option, the following warning will be displayed:

```
Warning: Changing the size to [extended] requires the use of increased
255 chars long ifAlias object of ifXtable from IF-MIB(RFC 2233)
```

You can always configure a 255-character-long string regardless of the configured value of ifAlias size. Its value only affects the SNMP behavior.

Port Isolation

The Port Isolation feature blocks accidental and intentional inter-communication between different customers residing on different physical ports. This feature provides a much simpler blocking mechanism without the use of *ACL* hardware. The fundamental requirements are as follows:

- Blocking Rules: All traffic types received on a isolation port is blocked from being forwarded through other 'isolation' ports.
- All traffic types received on an isolation port can be forwarded to any other port.
- All traffic types received on non-isolation ports are permitted to be forwarded to isolation ports.

There is no access-list hardware use. The blocking mechanism is a set of one or two table memories. These resources are not shared with other features, nor do they have any scaling limits that can be reached by configuring this feature. Port isolation can be configured in conjunction with other features, including VPLS, IDM, and *Extreme Network Virtualization (XNV)*. However, you cannot configure a mirror-to port to be an isolated port.

Configuring Port Isolation

To enable isolation mode on a per-port basis, perform the following tasks:

1. Issue the `configure ports` *`port_list`* **`isolation`** `[`**`on`** `|` **`off`**`]` command.

   You can issue this command either on a single port or on a master port of a load share group. If you issue this command on a non-master port of a load share group, the command will fail. When a port load share group is formed, all of the member ports assume the same isolation setting as the master port.
2. Issue the `show port info detail` and `show port info` outputs to display the current isolation settings.

Energy Efficient Ethernet

As part of the 802.3az standard, Energy Efficient Ethernet (EEE) is targeted at saving energy in Ethernet networks for a select group of physical layer devices, or PHYs. The PHYs use EEE during idle periods to reduce power consumption. If you do not utilize EEE, the PHY is fully powered up even when there is no traffic being sent. Enabling

EEE reduces significant power consumption on the switch. Within ExtremeXOS, a PHY/switch combination, or a PHY with autogrEEEN capability is provided to allow EEE to work. In a typical setup, the PHY and switch communicate when to enter or exit low power idle (LPI) mode.

AutoGrEEEn technology implements the EEE standard directly into PHYs, and enables the EEE mode when interfacing with non-EEE enabled MAC devices,. This allows you to make existing network equipment EEE-compliant by simply changing the PHY devices. EEE is currently only implemented for copper ports

Previously, most wireline communications protocols used continuous transmission, consuming power whether or not data was sent. EEE puts the PHY in an active mode only when real data is sent on the media. To save energy during gaps in the data stream, EEE uses a signaling protocol that allows a transmitter to indicate the data gap and to allow the link to go idle. This signaling protocol is also used to indicate that the link needs to resume after a pre-defined delay.

The EEE protocol uses an LPI signal that the transmitter sends to indicate that the link can go idle. After sending LPI for a period (Ts = time to sleep), the transmitter stops signaling altogether, and the link becomes quiescent. Periodically, the transmitter sends some signals so that the link does not remain quiescent for too long without a refresh. When the transmitter wishes to resume the fully functional link, it sends normal idle signals. After a pre-determined time (Tw = time to wake), the link is active and data transmission resumes.

The EEE protocol allows the link to be re-awakened at any time; there is no minimum or maximum sleep interval. This allows EEE to function effectively in the presence of unpredictable traffic. The default wake time is defined for each type of PHY, and is designed to be similar to the time taken to transmit a maximum length packet at the particular link speed.

The refresh signal serves the same purpose as the link pulse in traditional Ethernet. The heartbeat of the refresh signal ensures that both partners know that the link is present, and allows for immediate notification following a disconnection. The frequency of the refresh prevents any situation where one link partner is disconnected and another inserted without causing a link fail event. This maintains compatibility with security mechanisms that rely on continuous connectivity and require notification when a link is broken.

The maintenance of the link through refresh signals also allows higher layer applications to understand that the link is continuously present, preserving network stability. You can also use the refresh signal to test the channel, and create an opportunity for the receiver to adapt to changes in the channel characteristics. For high speed links, this is vital to support the rapid transition back to the full speed data transfer without sacrificing data integrity. The specific makeup of the refresh signal is designed for each PHY type to assist the adaptation for the medium supported.

Supported Platforms

EEE is supported on the following Extreme Networks platforms:

- 4120—copper 10/100/1000/2500 and multi-rate ports
- 4220—copper 10/100/1000 and multi-rate ports

  Note
  A 4220 Series hardware limitation prevents multi-rate ports from supporting EEE when they are operating at 2.5 Gbps or 5 Gbps speeds.

- 5120-24XT-4Y—1G/10GBase-T copper ports
- 5320—copper 10/100/1000 and multi-rate ports
- 5420—copper 10/100/1000 and multi-rate ports
- 5520—copper 10/100/1000 and multi-rate ports
- 5720—copper multi-rate ports

  Note
  A 5720 Series hardware limitation prevents multi-rate ports from supporting EEE when they are operating at 2.5 Gbps or 5 Gbps speeds.

- 7520-48XT—copper 1G/10G ports

Note
A hardware limitation prevents multi-rate ports from supporting EEE when they are operating at 2.5 Gbps or 5 Gbps speeds.

Configuring Energy Efficient Ethernet

- To enable or disable the EEE feature on EXOS, use the following command:

  `configure ports` *`port_list`* `eee [`**`on`**` | `**`off`**`]`

  The keyword on specifies that the port advertises to its link partner that it is EEE capable at certain speeds. If both sides, during auto-negotiation, determine that they both have EEE on and are compatible speed wise, they will determine other parameters (how long it takes to come out of sleep time, how long it takes to wake up) and the link comes up. During periods of non-activity, the link will shut down parts of the port to save energy. This is called LPI for low power idle. When one side sees it must send something, it wakes up the remote and then transmits.
- To display the statistics of the EEE features, use the following command:

  `show` **`ports`** *`port_list`* **`eee`** `{`**`no-refresh`**` | `**`refresh`**` } { `**`port-number`**`}`

Node Alias

Node Alias Overview

Node Alias is a feature that you can use to discover information about end systems on a per-port basis. As the packets from the end systems transit through a Node Alias-enabled port, its key attributes are captured in a database that can be queried

(see CTRON Alias MIB on page 1856). Enabling Node Alias on ports of interest provides a powerful troubleshooting tool for network administrators.

Note
By default, Node Alias is disabled on all ports.

Limitations

Currently, 100 packets per second per slot are admitted for snooping. Packets are dropped if the rate exceeds 100 pps.

Node Alias Database Information

Each entry in the Node Alias database consists of the following fields:

- Incoming port
- *VLAN* ID
- Source MAC address
- Protocol address—Depending on the protocol type, this could be the IPv4/IPv6 source IP address, or the UDP source port in cases where the destination IP is broadcast, etc., the ID and Priority for *VRRP*.
- Node Alias ID—A unique number that identifies an entry
- Entry creation time

The database is capable of holding 8,192 entries per-slot. To avoid starvation among ports, each port is allowed to create only its share of entries in the database. The default maximum per port is 8,192 divided by the number of ports on the slot. For example, on a 64-port slot, this translates to 128 entries per port. You can modify the per-port maximum (`configure` **`nodealias ports`** `[`*`port_list`* `|`**`all`**`]` **`maxentries`** *`entries`*) up to the limit of 8,192.

As a result of snooping one frame, Node Alias may create additional entries to facilitate the searching based on finer details, such as protocol type. For example, when a *BGP (Border Gateway Protocol)* frame is received, two entries are created: one entry with protocol type IP, and another entry with protocol type BGP.

Node Alias also supports VMAN. For VMANs, itt includes the VMAN ID, instead of VLAN ID in the database.

If the ports are combined to form a *LAG*, Node Alias must be enabled on individual ports; enabling Node Alias on the master port does not automatically enable Node Alias on member ports.

Supported Protocols

Node Alias is enabled by default to capture all of the following supported protocols: IPv4, IPv6, *OSPF*, BGP, VRRP, DHCPS, DHCPC, BOOTPS, BOOTPC, UDP, BPDU, LLMNR, SSDP, and mDNS. You can selectively disable any of these protocols.

Note

- ARP is categorized under IP.
- UDP entry is created when destination IP address is broadcast.
- BPDU means *STP* and GVRP frames.

Configuring Node Alias

Setting up Node Alias takes two steps:

1. Enabling (or disabling) Node Alias on port(s):

   Note
   By default, Node Alias is disabled on all ports.

   Note
   Node Alias should be separately enabled/disabled on a *LAG* ports.

   ```
   enable nodealias ports [port_list | all]

   disable nodealias ports [port_list | all]
   ```

2. Enabling (or disabling) certain protocols for Node Alias to snoop:

   Note
   By default, Node Alias is enabled to capture all of the following supported protocols: IPv4, IPv6, OSPF, *BGP*, *VRRP*, DHCPS, DHCPC, BOOTPS, BOOTPC, UDP, BPDU, LLMNR, SSDP, and mDNS.

   ```
   enable nodealias protocol [protocol_name | all]

   disable nodealias protocol [protocol_name | any]
   ```

To modify the per-port maximum number of alias entries in the Node Alias database, use the following command:

```
configure nodealias ports [port_list |all] maxentries entries
```

To show Node Alias information, use the following commands:

```
show nodealias

show nodealias ports {port_list| all}

show nodealias mac mac_address {protocol protocol_name | ports
[port_list | all]}
```

```
show nodealias ip ip_address

show nodealias protocol protocol_name
```

To clear node alias entries out of the Node Alias feature database by specified port(s) or alias ID, use the following command:

```
clear nodealias { ports [port_list | all] | alias-id alias_id }
```

IP Device Tracking

IP Device Tracking enables the switch to send periodic ARP probes to devices learned in the nodealias table on a per-port basis. Use this feature to keep NetLogin authentication sessions active for devices that generate minimal traffic and would otherwise be disconnected by idle timeout.

When a device responds to an ARP probe, the switch resets the NetLogin inactivity timer for that session. When a device fails to respond to a configurable number of consecutive probes, the switch marks that device Inactive. IP Device Tracking operates on ARP traffic only and applies to directly connected Layer 2 devices.

Note
IP Device Tracking must be configured independently on each LAG member port.

Device Status Values

Each nodealias entry on a port with IP Device Tracking configured is assigned one of the following status values:

- **Active**—The port has IP Device Tracking enabled and the device responded to an ARP probe.
- **Inactive**—The port has IP Device Tracking enabled and the device has not responded to consecutive ARP probes equal to the configured max-failures value.
- **Not Tracked**—IP Device Tracking is disabled on the port. No ARP probes are sent.

ARP Probe Parameters

The following parameters control ARP probe behavior and are configurable per port:

- **max-failures**—Number of consecutive probe failures before the device status changes from Active to Inactive. Range: 1–10. Default: 3.
- **interval**—Time in seconds between ARP probes. Range: 30–3600. Default: 60.
- **initial-delay**—Time in seconds the switch waits before sending the first ARP probe after a device is detected or a link comes up. A value of 0 causes probing to begin immediately. Range: 0–3600. Default: 10.
- **ipaddress**—Sender IP address used in ARP probe packets. Default: 0.0.0.0.

Note
ARP probe parameters can be configured whether IP Device Tracking is on or off, allowing parameters to be staged before the feature is enabled.

Enabling and Disabling IP Device Tracking

To enable or disable IP Device Tracking on one or more ports:

```
configure nodealias ports ip-device-tracking
```

Setting the port to `off` (default) stops ARP probing and sets all devices on that port to Not Tracked.

Configuring ARP Probe Parameters

To configure ARP probe parameters on one or more ports. Specify one or more parameters in any order in a single command:

```
configure nodealias ports ip-device-tracking arp-probe
```

The following example configures port 7 with a 50-second probe interval, 3-second initial delay, sender IP 20.0.0.1, and a maximum of 4 consecutive failures:

```
configure nodealias ports 7 ip-device-tracking arp-probe interval 50 initial-delay 3
ipaddress 20.0.0.1 max-failures 4
```

Displaying IP Device Tracking Status

To display the IP Device Tracking status of nodealias entries, optionally filtered by port:

```
show nodealias ip-device-tracking
Port   MAC Address          Alias ID  Time Last Learned    VID   Protocol  Source
IP        Status
-----  -----------------  ----------  -------------------  ----  --------
---------------  -----------
7      00:00:c0:01:0b:07          15  11-20-2025 16:59:45  10    ARP
20.0.0.15        Active
7      00:00:c0:01:0b:02          10  11-20-2025 16:59:13  10    ARP
20.0.0.10        Inactive
8      00:00:c0:01:0b:14          28  11-20-2025 17:00:14  10    ARP
20.0.0.28        Not Tracked
```

To display the ARP probe configuration for all ports, optionally filtered by port:

```
show nodealias ip-device-tracking configuration
Port    IP Device       Max     Probe  Initial  Probe
        Tracking   Failures  Interval    Delay  Sender IP
------  ---------  --------  --------  -------  ---------------
2       On                3        30       10  20.0.0.1
7       On                4        50        3  20.0.0.1
8       Off               3        60       10  0.0.0.0
```

Supported Platforms

All platforms.

Using Locally Administered MAC Addresses

ExtremeXOS switches do not use a unique per-port MAC address when transmitting bridge protocol data units (BPDUs). As a result, switch management can become inaccessible when switch MAC addresses are learned on the wrong L2 path (corresponding to a blocking port). Starting with ExtremeXOS 22.3, you can direct the

switch to generate locally administered addresses (LAA). When this feature is enabled, for Spanning Tree Protocol (STP), the source MAC address is changed to LAA MAC while transmitting the STP/RSTP/MSTP BPDUs. When LAA is enabled, the source MAC address is assigned as 06xx:xx:xx:xx;xx. The "06" replaces the first byte of the system MAC address.

Limitations

Only STP uses LAA MAC.

To direct the switch to generate unique locally administered MAC addresses, use the following command:

```
enable switch locally-administered-address
```

To disable generating unique locally administered MAC addresses, use the following command:

```
disable switchlocally-administered-address
```

To see the status of this feature, use the command:

```
show switch {detail}
```

Enabling/Disabling the USB Port

For enhanced security, you can enable or disable a USB port on a switch. The USB port is enabled by default.

Supported Platforms

All ExtremeSwitching platforms.

Limitations

Stack support is not available. You need to enable this command individually on each node in a stack.

Using USB Port Commands

To enable the USB port, use the following command:

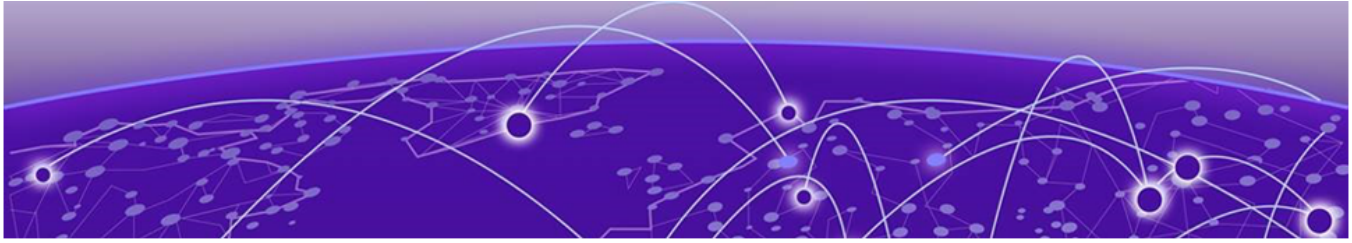
```
enable switch usb
```

To disable the USB port, use the following command:

```
disable switch usb
```

To show the status of the USB port, use the following command:

```
show switch usb
```



ACLs

ACLs Overview on page 341
Two-Stage ACL on page 342
ACL Rule Syntax on page 346
Layer-2 Protocol Tunneling ACLs on page 368
ACL Byte Counters on page 369
Dynamic ACLs on page 370
CCOS ACL Match Criteria on page 382
ACL Evaluation Precedence on page 382
Applying ACL Policy Files on page 384
ACL Mechanisms on page 388
Policy-Based Routing on page 402
ACL Troubleshooting on page 409

This chapter discusses *ACL (Access Control List)*s.

ACLs Overview

An *ACL* is used to define packet filtering and forwarding rules for traffic traversing the switch. Each packet arriving on an ingress port and/or *VLAN (Virtual LAN)* is compared to the access list applied to that interface and is either permitted or denied. Packets egressing an interface can also be filtered on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document. However, only a subset of the filtering conditions available for ingress filtering are available for egress filtering.

In addition to forwarding or dropping packets that match an ACL, the switch can also perform additional operations such as incrementing counters, logging packet headers, mirroring traffic to a monitor port, sending the packet to a *QoS (Quality of Service)* profile, and metering the packets matching the ACL to control bandwidth. (Metering is supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.) Using ACLs has no impact on switch performance (with the minor exception of the mirror-cpu action modifier).

ACLs are typically applied to traffic that crosses Layer 3 router boundaries, but it is possible to use access lists within a Layer 2 virtual LAN (VLAN).

ACLs in ExtremeXOS apply to all traffic. For example, if you deny all the traffic to a port, no traffic, including control packets, such as *OSPF (Open Shortest Path First)* or

RIP (Routing Information Protocol), will reach the switch and the adjacency will be dropped.

Note
Some locally CPU-generated packets are not subject to egress ACL processing.

You must explicitly allow those types of packets (if desired).

ACLs are created in two different ways. One method is to create an ACL policy file and apply that ACL policy file to a list of ports, a VLAN, or to all interfaces. The second method to create an ACL is to use the CLI to specify a single rule, called a dynamic ACL; this is the default.

Note
ACLs applied to a VLAN are actually applied to all ports on the switch, without regard to VLAN membership. The result is that resources are consumed per chip.

An ACL policy file is a text file that contains one or more ACL rule entries. This first method creates ACLs that are persistent across switch reboots, can contain a large number of rule entries, and are all applied at the same time. See ACL Rule Syntax on page 346 for information about creating ACL rule entries.

Policy files are also used to define routing policies. Routing policies are used to control the advertisement or recognition of routes communicated by routing protocols. ACL policy files and routing policy files are both handled by the policy manager, and the syntax for both types of files is checked by the policy manager.

Note
Although ExtremeXOS does not prohibit mixing ACL and routing type entries in a policy file, it is strongly recommended that you do not mix the entries and do use separate policy files for ACL and routing policies.

Dynamic ACLs persist across reboots; however, you can configure non-persistent dynamic ACLS that disappear when the switch reboots. Dynamic ACLs consist of only a single rule. Multiple dynamic ACLs can be applied to an interface. See Layer-2 Protocol Tunneling ACLs on page 368 for information about creating dynamic ACLs. The precedence of ACLs can be configured by defining zones and configuring the priority of both the zones and the ACLs within a zone. See Configuring ACL Priority for more information.

Two-Stage ACL

The following diagram shows the three *ACL* processors available that can be used for filtering the packets at various stages of processing:

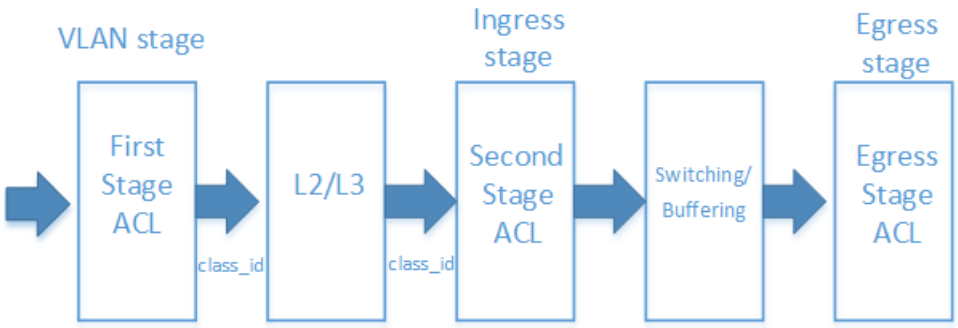


Figure 24: ACL Stages

First Stage ACL / *VLAN* Processor: is used to filter packets before ingress processing. It can be used to assign the VLAN, set a CLASS ID, or perform other more traditional ACL actions, such as drop or count. In general, this stage's scale, actions, and match criteria are more limited than the ingress stage. However, the high-level architecture of the first stage ACL is the same as the second stage ACL in that it is composed of a series of slices, or individual TCAM elements. First stage ACL rules are included in the policy file or dynamic ACLs in the same way as regular second stage ACL rules. To specify that a rule is to be added to the first stage ACL table, use the "class-id <class-id>" action.

Second Stage ACL / Ingress Filter Processor: is used to filter packets for ingress processing and is the primary hardware resource used for ingress user ACLs. While this stage follows the L2 and L3 lookups, the packet data presented to this stage is pre-modified, except in the case of tunneling. In general, this stage is the most capable and scalable of the 3 stages. Second stage ACL rules can additionally match the class-id specified as an action in a first stage ACL rule. This is done by listing "class-id <class-id>" in the match clause of the rule.

Egress ACL: is used to filter egressing packets after packet switching, queuing, and buffering operations have been performed. It can be used to deny or modify (e.g. 802.1p, DSCP) packets but cannot be used to redirect or change *QoS*. In general, this stage's scale, actions, and match criteria are more limited than the ingress stage.

The two-stage ingress classification is supported for static user ACL policies, dynamic CLI-driven user ACLs, and dynamic API-driven ACLs.

Platform Support

This feature is supported on all Universal platforms.

Feature Description

Rules in the first classifier are set up with an action to set class_id. Rules in the second classifier are setup to use the class_id as the key to match on the identity specific policies. The class_id is the common attribute between the two classifiers/ tables, uniquely identifies the role of the identity.

This feature introduces one new *ACL* action modifier for specifying the class-id from the first stage that will be input into the second stage. It also introduces one new ACL match criteria for matching the class-id within the second stage.

When a rule is installed in the first stage ACL table, it will be accounted for in the "Stage: LOOKUP" section of `show access-list usage acl-slice port` *port* . When a rule is installed in the second stage ACL table, it is accounted for in the "Stage: INGRESS" section of this command. For example:

```
# show access-list usage acl-slice port 1
Ports 1-54
Stage: INGRESS
Slices:          Used: 0  Available: 16
Virtual Slice  * (physical slice  0) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  1) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  2) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  3) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  4) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  5) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  6) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  7) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  8) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  9) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice 10) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice 11) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice 12) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice 13) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice 14) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice 15) Rules:   Used:      0  Available:    256
Stage: EGRESS
Slices:          Used: 0  Available: 4
Virtual Slice  * (physical slice  0) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  1) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  2) Rules:   Used:      0  Available:    256
Virtual Slice  * (physical slice  3) Rules:   Used:      0  Available:    256
Stage: LOOKUP
Slices:          Used: 0  Available: 4
Virtual Slice  * (physical slice  0) Rules:   Used:      0  Available:    512
Virtual Slice  * (physical slice  1) Rules:   Used:      0  Available:    512
Virtual Slice  * (physical slice  2) Rules:   Used:      0  Available:    512
Virtual Slice  * (physical slice  3) Rules:   Used:      0  Available:    512
Stage: EXTERNAL

Virtual Slice :  (*) Physical slice not allocated to any virtual slice.
5420F-48P-4XE.10 #
```

Limitations

- The second stage ACL will always override any qosprofile set in the First stage.
- A first stage ACL rule will not work for untagged traffic when "vlan-id" is used as a matching condition or when applied to a vlan .

- Matching "arp-sender-address" OR "arp-target-address" in the first stage ACL is not supported. However, matching both conditions is supported on select platforms.
- L4 port ranges are not supported in the first stage ACL.

Table 24: First-Stage ACL Support Actions

Platform	Permit	Deny	Count	Replace-dot1p-value	qosprofile	Replace-dot1p
4120	Y	Y	Y	Y	Y	Y
4220	Y	Y	Y	Y	Y	Y
5120	Y	Y	Y	Y	Y	Y
5320	Y	Y	Y	Y	Y	Y
7520	Y	Y	Y	Y	Y	Y
7720	Y	Y	Y	Y	Y	Y

Two-Stage Policy Example

The following example policy demonstrates how these new tokens can be used to create “user profiles” where each user is identified by source MAC address:

```
twostage_example1.pol:
 # First stage rules:

 entry firststage_1 {
if{
    ethernet-source-address 00:00:00:00:00:01;
} then {
    class-id 7;
}}
entry firststage_2 {
if {
    ethernet-source-address 00:00:00:00:00:02;
} then {
    class-id 8;
}}entry firststage_3 {
if {
    ethernet-source-address 00:00:00:00:00:03;
} then {
    class-id 7;
}}

 # Second stage rules:

entry secondstage_1 {
 if{
   class-id 7;
   destination-address 10.68.9.0/24;
} then {
    permit;
}}

entry secondstage_2 {
if {
    class-id 8;
    destination-address 10.68.0.0/16;
```

```
} then {
    permit;
}}entry secondstage_3 {
if {
} then
{entry permit_arp {
if {
ethernet-type 0x0806;
} then {
permit;
}
}
   deny;
    }}
```

The above example policy would have the following resulting behavior:

1. MAC addresses 00:00:00:00:00:01 and 00:00:00:00:00:03 would be permitted to access subnet 10.68.9.0/24
2. MAC address 00:00:00:00:00:02 would be permitted to access subnet 10.68.0.0/16
3. All other traffic would be dropped.

ACL Rule Syntax

An *ACL* rule entry consists of:

- A rule entry name, unique within the same ACL policy file or among Dynamic ACLs.
- Zero or more match conditions.
- Zero or one action (permit or deny). If no action is specified, the packet is permitted by default.
- Zero or more action modifiers.

Each rule entry uses the following syntax:

```
entry <ACLrulename>{
    if  {
        <match-conditions>;
    } then {
        <action>;
        <action-modifiers>;
    }
}
```

The following is an example of a rule entry:

```
entry  udpacl {
    if  {
        source-address 10.203.134.0/24;
        destination-address 140.158.18.16/32;
        protocol  udp;
        source-port 190;
        destination-port  1200 - 1250;
    } then {
        permit;
    }
}
```

An ACL rule is evaluated as follows:

- If the packet matches all the match conditions, the action and any action modifiers in the then statement are taken.
- For ingress ACLs, if a rule entry does not contain any match condition, the packet is considered to match and the action and any action modifiers in the rule entry's "then" statement are taken. For egress ACLs, if a rule entry does not contain any match condition, no packets will match. See Matching All Egress Packets on page 347 for more information.
- If the packet matches all the match conditions, and if there is no action specified in the then statement, the action permit is taken by default.
- If the packet does not match all the match conditions, the action in the then statement is ignored.

Matching All Egress Packets

Unlike ingress *ACL*s, for egress ACLs you must specify either a source or destination address, instead of writing a rule with no match conditions.

For example, an ingress ACL deny all rule could be:

```
entry DenyAllIngress{
    if {
    } then {
        deny;
    }
}
```

The previous rule would not work as an egress ACL.

The following is an example of an egress ACL deny all rule:

```
entry DenyAllEgress{
    if {
        source-address 0.0.0.0/0;
    } then {
        deny;
    }
}
```

Comments and Descriptions in ACL Policy Files

In *ACL* policy files, there are two types of textual additions that have no effect on the ACL actions: comments and descriptions. A comment is ignored by the policy manager and resides only in the policy file. Comments are not saved in the switch configuration and are not displayed by the show policy command. A description is saved in the policy manager and is displayed when the ACL is displayed.

- You can display the ACL using the following two commands:

  `show policy {`*`policy-name`* `|` **`detail`**`}`

  `show access-list {`**`any`** `|` **`ports`** *`port_list`* `|` **`vlan`** *`vlan_name`*`} {`**`ingress`** `|` **`egress`**`}`

For example, the following policy, saved in the file denyping.pol, contains both a comment and a description:

```
# this line is a comment
@description "This line is a description for the denyping.pol"
entry ping_deny_echo-request {
    if {
        protocol icmp;
        icmp-type echo-request;
    } then {
        deny;
        count pingcount_deny;
    }
}
```

Note that the description begins with the tag @description and is a text string enclosed in quotes.

- You can apply the policy to port 1 using the following command:

```
configure access-list denyping port 1
```

- and display the policy using the following command:

```
show policy denyping
```

The output of this command is similar to the following:

```
Policies at Policy Server:
Policy: denyping
@description This line is a description for the denyping.pol
entry ping_deny_echo-request {
if match all {
protocol icmp ;
icmp-type echo-request ;
}
Then {
deny  ;
count pingcount_deny ;
}
}
Number of clients bound to policy: 1
Client: acl bound once
```

Types of Rule Entries

In ExtremeXOS, each rule can be one of the following types:

- L2 rule—A rule containing only Layer 2 (L2) matching conditions, such as Ethernet MAC address and Ethernet type.
- L3 rule—A rule containing only Layer 3 (L3) matching conditions, such as source or destination IP address and protocol.
- L4 rule—A rule containing both Layer 3 (L3) and Layer 4 (L4) matching conditions, such as TCP/UDP port number.

Match Conditions

You can specify multiple, single, or zero match conditions. If you do not specify a match condition, all packets match the rule entry. Commonly used match conditions are:

- `ethernet-source-address [mac-address | pre-defined-mac ] mask`—Ethernet source address
- `ethernet-destination-address [mac-address | pre-defined-mac ] mask`—Ethernet destination address and mask
- `ethernet-type value {mask value}`—Ethernet type, accepts an optional mask.
- `source-address prefix`—IP source address and mask
- `destination-address prefix`—IP destination address and mask
- `destination-port value {mask value}`—IP destination port, accepts optional mask
- `source-port [value {mask value}|range]`—TCP or UDP source port with optional mask or TCP or UDP source port range
- `destination-port [port {mask value} |range]`—TCP or UDP destination port with optional mask or TCP or UDP destination port range
- `ttl value {mask value}`—condition with optional mask that matches IPv4 Time-To-Live and IPv6 Hop Limit.
- `ip-tos value {mask value}`—this condition accepts optional masks
- `vlan-format`—matches packets based on their *VLAN* format. Can be one of the following values:
  - untagged—all untagged packets
  - single-tagged—all packets with only a single tag
  - double-tagged—all packets with a double tag
  - outer-tagged—all packets with at least one tag; for example, single tag or double tag

Note

ACL rules using the `vlan-format` match condition can only be applied to a physical port binding point. Applying such a rule to the `any` or `vlan` binding point returns the following error:

```
Error: ACL install operation failed - internal failure
```

To apply a `vlan-format` rule to the `any` or `vlan` binding point, add a `source-physical-port` qualifier to the same policy entry:

```
entry one {
    if {
        vlan-format outer-tagged;
        source-physical-port 2;
    } then {
        count c1;
    }
}
```

- `fragments`—matches any fragment of fragmented packet, including the first fragment

- `first-fragments`—matches only the first fragment of a fragmented packet.
- `l4-match` *`value`* **`offset`** *`offset`* **`mask`** *`mask value`*—generic bit-matching pattern starting at the Layer 4 header of four separate chunks of 32-bits, each fully bit-maskable with a unique offset. Unlike others, this match criteria can appear up to four times in a single rule, each specified as a logical AND, to match up to four separate chunks of 32-bits. Each chunk is fully bit-maskable with a unique offset. The matching data must be within the first 128 bytes of the packet. This match criteria is intended for advanced users only.
  - *`value`*—32-bit value
  - *`offset`*—number of bytes from the start of the Layer 4 header (for example, TCP header)
  - *`mask`*—32-bit mask value applied to *`value`* for matching. Mask is optional. The default is 0xffffffff.

Table 25 on page 356 describes all the possible match conditions.

Actions

The actions are:

- `permit`—The packet is forwarded.
- `deny`—The packet is dropped.

The default action is permit, so if no action is specified in a rule entry, the packet is forwarded.

The following actions are supported on all platforms:

- `deny-cpu`—Prevents packets that are copied or switched to the CPU from reaching the CPU. The data-plane forwarding of these packets is unaffected. For example, use this action to match broadcast packets and prevent them from reaching the CPU, but still allow them to be flooded to other *VLAN* members. You can also use this action to match Spanning Tree Protocol packets and prevent them from reaching the CPU, and instead flood them to other VLAN members in certain configurations where Spanning Tree is enabled.
- `copy-cpu-off`—Prevents packets that are copied to the CPU from reaching the CPU. The data-plane forwarding of these packets is unaffected. For example, use this action to cancel the “mirror-cpu” action in another rule. This action does not prevent packets that are switched to the CPU (for example, broadcast, layer-3 unicast miss) from reaching the CPU.
- `copy-cpu-and-drop`—Overrides the above action to facilitate the above action in a “catch-all” rule. It sends matching packets only to the CPU.
- `add-vlan-id`—Adds a new outer VLAN ID. If the packet is untagged, it adds a VLAN tag to the packet. If the packet is tagged, it adds an additional VLAN tag. Only supported in VLAN lookup stage (VFP).
- `replace-dscp-value`—Replaces the existing DSCP value of the packet

- `do-ipfix`—Records the matching packet. Can be used on both ingress and egress. Attempting to install a policy with this action on an unsupported chip will result in failure in HAL.
- `do-not-ipfix`—Cancels recording for the matching packet. Can be used to reduce demand on egress IPFIX capacity (and to reduce recording loss) during packet flooding situations. Attempting to install a policy with this action on an unsupported chip will result in failure in HAL.
- `redirect-port-copy-cpu-allowed`—Redirects a packet out of an output port, but does not enforce a requirement that Copy to CPU must be cancelled.
- `redirect-port-list-copy-cpu-allowed`—Redirects a packet out of an output port to a list of ports, but does not enforce a requirement that Copy to CPU must be cancelled.

Action Modifiers

Additional actions can also be specified, independent of whether the packet is dropped or forwarded. These additional actions are called action modifiers. Not all action modifiers are available on all switches, and not all are available for both ingress and egress ACLs. The action modifiers are:

- `class-id` *`value 0-4095`*—Signifies that the rule will be installed in the LOOKUP stage access-list resource. Class-id range varies from platform to platform.
- `count` *`countername`*—Increments the counter named in the action modifier.

  Note
  The CLEAR-Flow counters work when the ACL is applied to a VLAN, but not if applied to a port or wildcard.

  - ingress— All platforms
  - egress— All platform. On egress, count does not work in combination with deny action.

  Note
  On egress, count does not work in combination with deny action in some platforms

- `add-vlan-id`—Adds a new outer *VLAN* ID. If the packet is untagged it will add a VLAN tag to the packet. If the packet is tagged, it will add an additional VLAN tag. Only supported in VLAN Lookup stage (VFP).
- `byte-count` *`byte counter name`*—Increments the byte counter named in the action modifier
- `packet-count` *`packet counter name`*—Increments the packet counter named in the action modifier.
- `log { rate-limit` *`burst_size`* `in` *`time_interval`* `}`—Logs the packet header and optionally rate limit the logs.
- `log-raw { rate-limit` *`burst_size`* `in` *`time_interval`* `}`—Logs the packet header in hex format and optionally rate limit the logs.

- `meter metername`—Takes action depending on the traffic rate. (Ingress and egress meters are supported on the platforms listed for these features in the Switch Engine v33.6.1 Licensing Guide document.
- `mirror`—Rules that contain mirror as an action modifier will use a separate slice.
- `mirror-cpu`—Mirrors a copy of the packet to the CPU in order to log it. It is supported only in ingress.
- `qosprofile qosprofilename`—Forwards the packet to the specified *QoS* profile.
  - ingress—all platforms
  - egress—does not forward the packets to the specified qosprofile. If the action modifier “replace-dot1p” is present in the ACL rule, the dot1p field in the packet is replaced with the value from associated qosprofile. All ExtremeSwitching Universal switches.
- `redirect ipv4 addr`—Forwards the packet to the specified IPv4 address.
- `redirect-no-replace-l2-sa IP nexthop address`—Forwards the packet to the specified IPv4 address without changing the source MAC address. Only apply to “L3 routable” traffic. Layer-2 traffic is not subject to matching.
- `redirect-port port`—Overrides the forwarding decision and changes the egress port used. If the specified port is part of a load share group then this action will apply the load sharing algorithm.
- `redirect-port-list port_list`—Supports multiple redirect ports as arguments. When used in an ACL, matching packets are now redirected to multiple ports as specified in the ACL while overriding the default forwarding decision. Maximum number of ports that can be mentioned in this list is 64.
- `redirect-port-no-sharing port`—Overrides the forwarding decision and changes the egress port used. If the specified port is part of a load share group then this action overrides the load sharing algorithm and directs matching packets to only this port.
- `redirect-name name`—Specifies the name of the flow-redirect that must be used to redirect matching traffic.
- redirect-vlan—Redirects the traffic to all ports in the matching VLAN. With L3 unicast routing, floods on the egress VLAN members.
- `replace-dscp`—Replaces the packet’s DSCP field with the value from the associated QoS profile.
  - ingress
  - egress
- `replace-dot1p`—Replaces the packet’s 802.1p field with the value from the associated QoS profile.
  - ingress
  - egress
- `replace-dot1p-value value`—Replaces the packet's 802.1p field with the value specified without affecting the QoS profile assignment.
  - ingress
  - egress

- `replace-ethernet-destination-address` *`mac-address`*—Replaces the packet's destination MAC address; this is applicable only to layer-2 forwarded traffic.
- `replace-vlan-id` —Replaces an outer VLAN ID in a double-tagged packet or a single VLAN tag in a single-tagged packet.
  - ingress
  - egress

Counting Packets and Bytes

When the *ACL* entry match conditions are met, the specified counter is incremented.

- The counter value can be displayed by the command:

  `show access-list counter {`*`countername`*`} {`**`any`** `|` **`ports`** *`port_list`* `|` **`vlan`** *`vlan_name`*`} {`**`ingress`** `|` **`egress`**`}`

  ExtremeSwitching series switches can use ACL byte counters as an alternative to ACL packet counters. See ACL Byte Counters on page 369 for more information.

  Note
  The maximum number of packets that can be counted with token packet-count or count is 18,446,744,073,709,551,616. The maximum number of bytes that can be counted with byte-count is also 18,446,744,073,709,551,616, which is equivalent to 288,230,376,151,711,744 packets that are sized at 64 bytes.

  Note
  Each packet increments only one counter in the egress direction. When there are multiple ACLs with action "count" applied in the port, only single counter based on the slice priority work.

Logging Packets

Packets are logged only when they go to the CPU, so packets in the fastpath are not automatically logged. You must use both the mirror-cpu action modifier and the log or log-raw action modifier if you want to log both slowpath and fastpath packets that match the *ACL* rule entry. Additionally, Kern.Info messages (or Kern.Card.Info on SummitStack) are not logged by default. You must configure an EMS filter to log these messages, for example, `configure log filter DefaultFilter add event kern.info`. See the Status Monitoring and Statistics chapter for information about configuring EMS.

The `log` and `log-raw` action modifiers can be rate-limited at both the global level and individually per rule. The global rate limit applies to all logs generated by `log` and `log-raw` actions across all ACL rules and has a default value of 10 logs per 5 seconds. Use the following command to configure the global rate limit:

```
configure access-list action-modifier log rate-limit <burst_size> in
<time_interval>
```

For example, to limit the total logging rate to 100 logs per hour, use the following configuration:

```
configure access-list action-modifier log rate-limit 100 in 3600
```

The *time_interval* parameter is specified in seconds, so 3600 seconds equals one hour. Individual `log` and `log-raw` actions within a rule can also have their own rate limits to prevent high-frequency rules from monopolizing the logging facility. When both global and rule-level rate limits are configured, logs must pass through both rate limiters before being sent to the logging facility. Rule-level rate limits prevent individual rules from consuming excessive logging capacity, while the global rate limit controls the total logging rate across all rules. See ACL Rule Syntax: Action Modifiers for information on configuring rate limits for individual rules.

Metering Packets

The `meter` *metername* action modifier applies a meter to the traffic defined by an *ACL*. For more information, see Meters on page 1469.

Mirroring Packets

You must enable port-mirroring on your switch. If you attempt to apply a policy that requires port-mirroring, the mirror action will be disabled until you enable the port-mirroring.

On ExtremeSwitching 5420 and 5520 series switches, mirroring can be configured on the same port as egress ACLs. Mirroring can send packets to port x and you can install your rule at egress port x, and the rule should match your mirrored traffic.

Starting with ExtremeXOS 30.1, a single ACL can have up to four mirror actions (each with a different mirror instance) to accomplish mirroring the same packet to multiple destinations (port, ERSPAN/remote-ip, RSPAN/remote-tag, etc).

Redirecting Packets

Packets are forwarded to the IPv4 address specified, without modifying the IP header (except the TTL is decremented and the IP checksum is updated). The IPv4 address must be in the IP ARP cache, otherwise the packet is forwarded normally. Only fast path traffic can be redirected. This capability can be used to implement Policy-Based Routing.

You may want to create a static ARP entry for the redirection IP address, so that there will always be a cache entry. See Policy-Based Routing on page 402 for more information.

Replacing DSCP or 802.1p Fields

Specify a *QoS* profile for matching packets. The field values are replaced with the value associated with that profile. In the following example, DiffServ replacement is

configured such that QP8 is mapped to code point 56. Matching packets are sent to QP8, and the DSCP value in the packet is set to 56.

```
entry voice_entry {
if {
source-address 2.2.2.2/32;
} then {
qosprofile qp8;
replace-dscp;
}
}
```

See Quality of Service on page 1453 for more details about QoS profiles, and 802.1p and DSCP replacement.

ACL Rule Syntax Details

The following table lists the match conditions that can be used with *ACL*s, and whether the condition can be used for ingress ACLs only, or with both ingress and egress. The conditions are not case-sensitive; for example, the match condition listed in the table as TCP-flags can also be written as tcp-flags. Within Table 25 are five different data types used in matching packets.

Table 25 lists general match conditions that apply to all traffic, unless otherwise noted. Table 26 on page 366 the data types and details on using them.

Table 25: ACL Match Conditions

Match Conditions	Description	Applicable IP Protocols/ Direction
`ethernet-type` *`number`*	Ethernet packet type. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): ETHER-P-IP (0x0800), ETHER-P-IPV6 (0x86DD).	Ethernet/ Ingress and Egress
[**`ethernet-source-address`** | **`ethernet-destination-address`**][*`mac-address`* | *`pre-defined-mac`*] { [**`mask`** | /] *`mask`* }	Match Ethernet source MAC address or Ethernet destination MAC address. The mask is optional, and is in the same format as the MAC address for example: `ethernet-source-address 00:01:02:03:01:01 mask ff:ff:ff:ff:00:00 or ethernet-source-address 00:01:02:03:01:01 / ff:ff:ff:ff:00:00` *`Pre-defined-mac`* can be any one of the following: `cdp-mac, csp-mac (Port Extender Control and Status Protocol), eaps-flush-fdb-mac, eaps-mac, eaps-ver1-mac, eaps-ver2-mac, edp-mac, elrp-dst-mac, elrpsrc-mac, elsm-mac, erps-mac, esrp-mac, lacp-mac, lldp-mac, msrp-mac, pvstp-mac, stp-mac, system-lla-mac and system-mac` Example with pre-defined-mac: `ethernet-source-address edp-mac mask ff:ff:ff:ff:00:00 or ethernet-source-address edp-mac / ff:ff:ff:ff:00:00` Only those bits of the MAC address whose corresponding bit in the mask is set to 1 will be used as match criteria. Therefore, the preceding example matches 00:01:02:03:xx:xx. If the mask is not supplied, it is assumed to be ff:ff:ff:ff:ff:ff. In other words, all bits of the MAC address are used for matching.	Ethernet/ Ingress and Egress
`source-address` *`prefix`*	IP source address and mask. Use either all IPv4 or all IPv6 addresses in an ACL. Using arbitrary mask arguments is supported on switches. Masks are not restricted to be of a subnet type. Examples of arbitrary IPv4 and IPv6 masks include 10.22.3.4 and 1:0:0:ffff:2:4. The 1s in the mask indicate the corresponding bits of the source-address that should be used as part of the match criteria.	All IP/ Ingress and Egress

[1] However, packets using the Ethernet type for VMANs, 0x88a8 by default, are handled by VMAN ACLs.

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`destination-address` *prefix*	IP destination address and mask. Using arbitrary mask arguments is supported on switches. Masks are not restricted to be of a subnet type. Examples of arbitrary IPv4 and IPv6 masks include 10.22.3.4 and 1:0:0:ffff:2:4. The 1s in the mask indicate the corresponding bits of the destination-address that should be used as part of the match criteria.	All IP/ Ingress and Egress
`source-port` *number*	TCP or UDP source port. You must also specify the protocol match condition to determine which protocol is being used on the port, any time you use the this match condition. In place of the numeric value, you can specify one of the text synonyms listed under destination port. If no source-port is specified, the default source-port is “any.”	TCP, UDP/ Ingress and Egress
`source-port` *range*	TCP or UDP source port. You must also specify the protocol match condition to determine which protocol is being used on the port, any time you use the this match condition. In place of the numeric value, you can specify one of the text synonyms listed under destination port. If no source-port is specified, the default source-port is “any.”	TCP, UDP/ Ingress
`source-port number { mask` *value* `}`	TCP or UDP port and mask. The mask is optional, and it can be decimal value or a hexadecimal value.	TCP,UDP/ Ingress and Egress

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`destination-port` *`number`*	TCP or UDP destination port. You must also specify the protocol match condition to determine which protocol is being used on the port, any time you use the this match condition. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): afs(1483), bgp(179), biff(512), bootpc(68), bootps(67), cmd(514), cvspserver(2401), dhcp(67), domain(53), eklogin(2105), ekshell(2106), exec(512), finger(79), ftp(21), ftp-data(20), http(80), https(443), ident(113), imap(143), kerberos-sec(88), klogin(543), kpasswd(761), krb-prop(754), krbupdate(760), kshell(544), ldap(389), login(513), mobileip-agent(434), mobileip-mn(435), msdp(639), netbios-dgm(138), netbios-ns(137), netbios-ssn(139), nfsd(2049), nntp(119), ntalk(518), ntp(123), pop3(110), pptp(1723), printer(515), radacct(1813), radius(1812), rip(520), rkinit(2108), smtp(25), snmp(161), snmptrap(162), snpp(444), socks(1080), ssh(22), sunrpc(111), syslog(514), tacacs-ds(65), talk(517), telnet(23), tftp(69), timed(525), who(513), xdmcp(177), zephyr-clt(2103), or zephyr-hm(2104).	TCP, UDP/ Ingress and Egress
`destination-port` *`range`*	TCP or UDP destination port. You must also specify the protocol match condition to determine which protocol is being used on the port, any time you use the this match condition. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): afs(1483), bgp(179), biff(512), bootpc(68), bootps(67), cmd(514), cvspserver(2401), dhcp(67), domain(53), eklogin(2105), ekshell(2106), exec(512), finger(79), ftp(21), ftp-data(20), http(80), https(443), ident(113), imap(143), kerberos-sec(88), klogin(543), kpasswd(761), krb-prop(754), krbupdate(760), kshell(544), ldap(389), login(513), mobileip-agent(434), mobileip-mn(435), msdp(639), netbios-dgm(138), netbios-ns(137), netbios-ssn(139), nfsd(2049), nntp(119), ntalk(518), ntp(123), pop3(110), pptp(1723), printer(515), radacct(1813), radius(1812), rip(520), rkinit(2108), smtp(25), snmp(161), snmptrap(162), snpp(444), socks(1080), ssh(22), sunrpc(111), syslog(514), tacacs-ds(65), talk(517), telnet(23), tftp(69), timed(525), who(513), xdmcp(177), zephyr-clt(2103), or zephyr-hm(2104).	TCP, UDP/ Ingress

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`destination-port number {`**`mask`** *`value`*`}`	TCP or UDP port and mask. The mask is optional, and it can be decimal value or a hexadecimal value. Only those bits of the destination-port whose corresponding bit in the mask is set to 1 will be used as match criteria.	TCP,UDP/ Ingress and Egress
TCP-flags *`bitfield`*	TCP flags. Normally, you specify this match in conjunction with the protocol match statement. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): ACK(0x10), FIN(0x01), PUSH(0x08), RST(0x04), SYN(0x02), URG(0x20), SYN_ACK(0x12).	TCP/Ingress and Egress
`IGMP-msg-type` *`number`*	IGMP message type. Possible values and text synonyms: v1-report(0x12), v2-report(0x16), v3-report(0x22), V2-leave (0x17), or query(0x11).	IGMP/ Ingress only
`ICMP-Type` *`number`*	ICMP type field. Normally, you specify this match in conjunction with the protocol match statement. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): echo-reply(0), echo-request(8), info-reply(16), info-request(15), mask-request(17), mask-reply(18), parameter-problem(12), redirect(5), router-advertisement(9), router-solicit(10), source-quench(4), time-exceeded(11), timestamp-request(13), timestamp-reply(14), v6-unreachable(1), v6-packet-too-big(2), v6-time-exceeded(3), v6-parameter-problem(4), v6-echo-request(128), v6-echo-reply(129), v6-mld-query(130), v6-mld-report(131), v6-mld-reduction(132), v6-router-solicitation(133), v6-router-advertisement(134), v6-neighbor-solicitation(135), v6-neighbor-advertisement(136), v6-redirect(137), v6-node-info-query(139), v6-node-info-reply(140).	ICMP/ Ingress only

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`ICMP-Code` *number*	ICMP code field. This value or keyword provides more specific information than the icmp-type. Because the value's meaning depends upon the associated icmp-type, you must specify the icmp-type along with the icmp-code (only available in IPv4). In place of the numeric value, you can specify one of the following text synonyms (the field values also listed); the keywords are grouped by the ICMP type with which they are associated: Parameter-problem: ip-header-bad(0), required-option-missing(1) Redirect: redirect-for-host (1), redirect-for-network (2), redirect-for-tos-and-host (3), redirect-for-tos-and-net (2) Time-exceeded: ttl-eq-zero-during-reassembly(1), ttl-eq-zero-during-transit(0) Unreachable: communication-prohibited-by-filtering(13), destination-host-prohibited(10), destination-host-unknown(7), destination-network-prohibited(9), destination-network-unknown(6), fragmentation-needed(4), host-precedence-violation(14), host-unreachable(1), host-unreachable-for-TOS(12), network-unreachable(0), network-unreachable-for-TOS(11), port-unreachable(3), precedence-cutoff-in-effect(15), protocol-unreachable(2), source-host-isolated(8), source-route-failed(5)	IPv4 only/ ICMP/ Ingress only
`source-sap`	SSAP is a 1 byte field with possible values 0-255 decimal. The value can be specified in decimal or hexadecimal. The SSAP field can be found at byte offset 15 in 802.3 SNAP and LLC formatted packets.	Ethernet/ Ingress Only
`destination-sap`	DSAP is a 1 byte field with possible values 0-255 decimal. The value can be specified in decimal or hexadecimal. The DSAP field can be found at byte offset 14 in 802.3 SNAP and LLC formatted packets. (Available on ExtremeSwitching series switches, SummitStack.)	Ethernet/ Ingress Only
`snap-type`	SNAP type is a 2 byte field with possible values 0-65535 decimal. The value can be specified in decimal or hexadecimal. The SNAP type field can be found a byte offset 20 in 802.3 SNAP formatted packets.	Ethernet/ Ingress Only

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`ttl number {mask value}`	Time To Live with mask.The mask is optional, and it can be decimal value or a hexadecimal value.Only those bits of the ttl whose corresponding bit in the mask is set to 1 will be used as match criteria.This can be used to match IPv4 Time-To-Live and IPv6 Hop Limit.	All IP/ Ingress and Egress.
`IP-TOS number`	IP TOS field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): minimize-delay 16 (0x10), maximize-reliability 4(0x04), minimize-cost2 (0x02), and normal-service 0(0x00).	All IP/ Ingress and Egress
`IP-TOS number {mask value}`	IP-TOS and mask.The mask is optional, and it can be decimal value or a hexadecimal value.Only those bits of the IP-TOS whose corresponding bit in the mask is set to 1 will be used as match criteria.	All IP/ Ingress and Egress
`dscp value`	DSCP field. In place of the value, you can specify one of the DSCP numeric values (for example, 8, 16, or 24).	All IP/ Ingress and Egress
`fragments`	IP fragmented packet including first fragment. FO = 0 (FO = Fragment Offset in IP header)	All IP, no L4 rules/ Ingress only
`first-fragments`	Matches only first fragmented packet. FO==0.	All IP/ Ingress only
`protocol number`	IP protocol field. For IPv6, this matches the Next Header field in the packet. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): egp(8), gre(47), icmp(1), igmp(2), ipip(4), Ipv6 over ipv4(41), ospf(89), pim(103), rsvp(46), st(5), tcp(6), or udp(17).	All IP/ Ingress and Egress
`vlan-format format`	VLAN-format matches packets based on its vlan format. Can be one of the 4 values: untagged - will match all untagged packets single-tagged - will match all packets with only single tag. double-tagged - will match all packets with double tag outer-tagged - will match all packets with at least one tag ex. single tag or double tag.	Ethernet/ Ingress and Egress

2 See the section Fragmented packet handling for details,
3 See the section IPv6 Traffic with L4 Match Conditions for details about specifying a protocol/port match with IPv6.

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`vlan-id number`	Matches the VLAN tag number or the VLAN ID which is given to a VLAN when created. The ACL rule can only be applied to ports or any, and not VLANs. The following restriction applies to all platforms: The vlan-id match condition matches on the “outer” tag of a VMAN.The vlan-id ACL keyword can be used in egress ACL.	Ethernet/ Ingress and Egress
`vlan-id number {mask value}`	VLAN-id and mask.The mask is optional, and it can be decimal value or a hexadecimal value.Only those bits of the Vlan tag Number or vlan id whose corresponding bit in the mask is set to 1 will be used as match criteria.	Ethernet/ Ingress and Egress
`vxlan-vni vni number`	VNI on an egress-terminated VXLAN packet (egress VTEP scenario) or on a transit switch. Available for both static and dynamic ingress ACLs.	Ethernet/ Ingress
`dot1p priority tag`	Creates an ACL with 802.1p match conditions, allowing the ACL to take action based on the VLAN tag priority. (Available on all platforms.)	All IP/ Ingress

`arp-sender-address prefix`
and
`arp-target-address prefix`
	Matches the ARP sender protocol address and target protocol address respectively.
	`prefix => IPv4 address / mask length`.
	They cannot be combined with an Ethernet-source-address or Ethernet-destination-address in the same rule.
	They can be used only when the ACL hardware database is configured to be internal for those platforms that support “external-table” ACL databases.
		ARP packets/ Ingress

ccos	Matches customer 802.1p on UNI or NNI ports. Available on all platforms. `ccos <ccos value>`	

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`cvid`	Use this match criteria in the following scenarios: Tagged VMAN ports: installing an ACL matching "cvid" on ingress or egress will match the inner vlan-id of a double tagged packet on a tagged VMAN port. Untagged VMAN ports: installing an ACL matching "cvid" on ingress or egress will match the single VLAN tag on an untagged VMAN port. CEP VMAN ports (with or without VPLS): installing an ACL matching "cvid" on ingress or egress will match the single VLAN tag on a CEP VMAN port (without translation). CEP VMAN ports with cvid translation (with or without translation): installing an ACL matching "cvid" on ingress will match the post-translation cvid. Installing an ACL matching "cvid" on egress will match the post-translation cvid.	Ethernet/ Ingress and Egress
`class-id`	This match condition can be specified on any rule within a policy file or within a list of dynamic access-lists. A rule cannot both match a class-id and specify a class-id as an action. When a "class-id" match criteria is specified, the associated rule will be programmed into the normal "INGRESS stage" access-list hardware resource. The range of valid class-id values varies per platform.	Ingress only.
unknown-l2-unicast	Matches the unknown L2 unicast packets	Ingress only
unknown-l2-multicast	Matches the unknown L2 multicast packets	Ingress only
unknown-l3-multicast	Matches the unknown L3 multicast packets	Ingress only
l2-da-hit	Matches the known L2 unicast packets	Ingress only
l4-match	Generic bit-matching pattern starting at the Layer 4 header of four separate chunks of 32-bits, each fully bit-maskable with a unique offset. Unlike others, this match criteria can appear up to four times in a single rule, each specified as a logical AND, to match up to four separate chunks of 32-bits. Each chunk is fully bit-maskable with a unique offset. The matching data must be within the first 128 bytes of the packet. This match criteria is intended for advanced users only.	Ingress only

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`ospf {msg-type} {version}`	Matches OSPF messages types and optionally OSPF version. If version is not specified, the rule matches both OSPFv2 and OSPFv3 packets. Valid OSPF message types are: *hello* *db-desc* *ls-request* *ls-update* *ls-ack* Valid OSPF version keywords are: *v2* *v3*	OSPF/ Ingress and Egress
`pim msg-type`	Matches PIM message types. Valid PIM message types are: *hello* *register* *register-stop* *join-prune* *boot-strap* *assert* *graft* *graft-ack* *crp-advt* *state-refresh* *df-election* *ecmp-redirect* *pim-flooding*	PIM / Ingress and Egress

Table 25: ACL Match Conditions (continued)

Match Conditions	Description	Applicable IP Protocols/ Direction
`packet-resolution` *`type`*	Valid packet types are: *`l2-broadcast`* *`known-l2-unicast`* *`unknown-l2-unicast`* *`known-l2-multicast`* *`unknown-l2-multicast`* *`known-l3-unicast`* *`unknown-l3-unicast`* *`known-l3-multicast`* *`unknown-l3-multicast`*	Ingress only
`packet-lookup-status` *`status1`* `{,` *`status2`* `{,` *`status3`*`}}`	Matches if the packet's lookup status satisfies all the statuses listed in the match condition. The lookup status value can be one of the following: • destination-mac-hit or destination-mac-miss • source-mac-miss or source-mac-hit or source-mac-move • source-mac-static For example: packet-lookup-status destination-mac-hit packet-lookup-status destination-mac-miss, source-mac-hit packet-lookup-status source-mac-move, source-mac-static	Ingress only

Note
When you use a configured ACL that contains a match condition with any MAC address, IGMP snooping stops working and IGMP joins are flooded to all ports in a VLAN. When you unconfigure the ACL, IGMP joins stop flooding.

Note
An ACL that matches the EAPS `ethernet-destination-address (00:e0:2b:00:00:04)` or `ethernet-source-address (00:e0:2b:00:00:01)` match condition with the permit action should not be applied to an EAPS master node on EAPS ring ports. Doing so causes an EAPS PDU loop. For the EAPS master node, you should use the `copy-cpu-and-drop` action with either of these match conditions.

Note
Directed ARP response packets cannot be blocked with ACLs from reaching the CPU and being learned.

Along with the data types described in Table 26, you can use the operators <, <=, >, and >= to specify match conditions. For example, the match condition, `source-port 190`, will match packets with a source port greater than 190. Be sure to use a space before and after an operator.

Note
These operators <,<=,> and >= are supported only in policy ACL mode and not in Dynamic ACL mode.

Table 26: ACL Match Condition Data Types

Condition Data Type	Description
prefix	IP source and destination address prefixes. To specify the address prefix, use the notation prefix/prefix-length. For a host address, prefix-length should be set to 32.
number	Numeric value, such as TCP or UDP source and destination port number, IP protocol number.
range	A range of numeric values. To specify the numeric range, use the notation: number - number
bit-field	Used to match specific bits in an IP packet, such as TCP flags and the fragment flag.
mac-address	6-byte hardware address.

IPv6 Traffic with L4 Match Conditions

If you apply an *ACL* policy using ACL that specifies L4 conditions, both IPv4 and IPv6 traffic will be matched. For example, the following ACL will match both IPv4 and IPv6 TCP packets that have their L4 destination port in the range of 120–150:

```
if {
protocol tcp;
destination-port 120 - 150;
}
Then {
permit;
count destIp;
}
}
```

Fragmented Packet Handling

Two keywords are used to support fragmentation in ACLs:

- **`fragments`**—FO field = 0 (FO means the fragment offset field in the IP header). This will match first fragment also (packets with FO = 0).
- **`first-fragments`**—FO == 0.

Policy file syntax checker

The **`fragments`** keyword cannot be used in a rule with L4 information. The syntax checker will reject such policy files.

The following rules are used to evaluate fragmented packets or rules that use the **fragments** or **first-fragments** keywords.

With no keyword specified, processing proceeds as follows:

- An L3-only rule that does not contain either the fragments or first-fragments keyword matches any IP packets.
- An L4 rule that does not contain either the fragments or first-fragments keyword matches non-fragmented or initial-fragment packets.

With the **fragments** keyword specified:

- An L3-only rule with the fragments keyword only matches fragmented packets.
- An L4 rule with the fragments keyword is not valid (see above).

With the **first-fragments** keyword specified:

- An L3-only rule with the first-fragments keyword matches initial fragment packets.
- An L4 rule with the first-fragments keyword matches initial fragment packets.

Wide Key ACLs

Wide key capability provides greater capacity for rules than single-wide key *ACL*s.

For example, a double-wide match key provides a 362-bit capacity, instead of a standard 181-bit single-wide key to be used with match conditions. A double-wide key allows you to add more match conditions to an ACL. It also allows matching on a full destination-source IPv6 address.

All switches

- Single-wide key
- Double-wide key

On all platforms, key width is configured manually (see Configuring Wide Key ACL Modes on page 368) and applies to all ACLs on the switch. An individual switch cannot be configured to operate in a mixed double- and single-wide mode. However, a SummitStack can have a mixture of modules and switches with some of them operating in a single-wide mode and some in a double-wide mode.

Double wide key ACLs allow additional condition combinations than single-wide ACLs. The existing supported condition combinations are described in Table 29 on page 395. The double-wide condition combinations that can be appended under the set union operation to the single-wide condition combinations are as follows:

- OVID, DIP, SIP, IpInfo(First-Fragments,Fragments), IP-Proto, DSCP, TCP-Flag, L4SP, L4DP
- SIPv6, IP-Proto, DSCP, TCP-Flag, L4SP, L4DP

For example, your single-wide mode supports condition combination A, B, and C, and the double-wide mode adds condition combinations D1 and D2. Then in a single-wide mode, the conditions of your rule should be a subset of either {A}, or {B}, or {C} and in a double-wide mode, the conditions of your rule should be a subset of either {A U D1}, or {A U D2}, or {B U D1}, or {B U D2}, or {C U D1}, or {C U D2}.

Limitations

The following are limitations associated with wide keys:

- Wide keys provides richer condition combinations. However, there is a tradeoff in ACL capacity; wide key ACLs consume more space.
- Only ingress ACLs support this feature. Egress and external ACLs do not support it.

Supported Platforms

Wide key ACLs are available on all ExtremeSwitching Universal platforms.

Configuring Wide Key ACL Modes

To configure the TCAM width of a switch (standalone or SummitStack), use the following command:

```
configure access-list width [double | single] [slot slotNo | all]
```

You must reboot for the configuration to take effect.

When switching from single-wide key mode to double-wide key mode and rebooting, the following conditions apply:

- Configurations that have less that one-half the number of *ACL* rules that the single-wide key mode supports, reboot successfully.
- Configurations that have more than one-half the number of ACL rules that the single-wide key mode supports, fail after the reboot.

When switching from double-wide key mode to single-wide key mode and rebooting, the following conditions apply:

- Configurations that do not use the additional condition combinations that double-wide key mode offers, reboot successfully.
- Configurations that use the additional condition combinations that the double-wide key mode offers, fail after the reboot.

To display the wide key mode settings, use the following command:

```
show access-list width [slot slotNo | all]
```

Layer-2 Protocol Tunneling ACLs

Three *ACL* match conditions and one ACL action interoperate with vendor-proprietary Layer-2 protocol tunneling on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

The following fields within 802.3 Subnetwork Access Protocol (SNAP) and LLC formatted packets can be matched:

- Destination service access point (SAP)
- Source SAP

The following field can be matched within Subnetwork Access Protocol (SNAP) packets only:

- SNAP type

The following ACL action is added to the specified switches:

- Replacement of the Ethernet MAC destination address

This action replaces the destination MAC address of any matching Layer-2 forwarded packets on the supported platforms. This action can be used to effectively tunnel protocol packets, such as *STP (Spanning Tree Protocol)*, across a network by replacing the well-known protocol MAC address with a different proprietary or otherwise unique MAC address. After tunnel egress, the MAC destination address can be reverted back to the well-known MAC address.

Note
The "replace-ethernet-destination-address" action applies only to Layer-2 forwarded packets.

ACL Byte Counters

An *ACL* byte counter associated with a particular rule, either dynamic or static, shows how many bytes of traffic have matched that ACL rule. You can use ACL byte counters as an alternative to packet counters on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document..

A new ACL action token has been added to associate a byte counter with an ACL, and a new corresponding token for a packet counter.

Following are the two new tokens:

`byte-count` *`byte counter name`*

`packet-count` *`packet counter name`*

An ACL rule specifying both packet and byte counter is rejected.

Below is an example of an ACL rule that uses a byte counter:

```
entry CountBytes {
    if {
        ethernet-source-address 00:aa:00:00:00:10;
    } then {
        byte-count CountBytes;
        permit;
    }
}
```

Below are two examples of ACL rules that use packet counters. The "packet-count" token is a synonym of the existing "count" token.

```
entry CountPacket1 {
    if {
        ethernet-source-address 00:aa:00:00:00:10;
```

```
    } then {
        count CountPacket1;
        permit;
    }
}

entry CountPacket2 {
    if {
        ethernet-source-address 00:aa:00:00:00:10;
    } then {
        packet-count CountPacket2;
        permit;
    }
}
```

The output of the `show access-list counter` and `show access-list dynamic counter` commands has been changed to include a new "Byte Count" column in addition to the "Packet Count" column. When a rule utilizes a byte counter, the "Byte Count" field is incremented and the "Packet Count" field stays at zero. If a rule utilizes a packet counter, the "Packet Count" field is incremented and the "Byte Count" field stays at zero.

Note
Byte counters and packet counters cannot be used at the same time in the same rule.

Dynamic ACLs

Dynamic *ACL*s are created using the CLI. They use a similar syntax and can accomplish the same actions as single rule entries used in ACL policy files. More than one dynamic ACL can be applied to an interface, and the precedence among the dynamic ACLs can be configured. By default, the priority among dynamic ACLs is established by the order in which they are configured.

Note
Dynamic ACLs have a higher precedence than ACLs applied using a policy file.

The steps involved in using a dynamic ACL on an interface are:

- Creating the Dynamic ACL Rule
- Configuring the ACL Rule on the Interface.
- Configuring ACL Priority
- Network-Zone Support in ACLs

Creating the Dynamic ACL Rule

Creating a dynamic *ACL* rule is similar to creating an ACL policy file rule entry. You specify the name of the dynamic ACL rule, the match conditions, and the actions and action-modifiers. You can configure a dynamic ACL to be persistent or non-persistent across system reboots. User-created access-list names are not case sensitive. The match conditions, actions, and action-modifiers are the same as those that are available for ACL policy files (see ACL Rule Syntax on page 346). In contrast to the ACL policy file

entries, dynamic ACLs are created directly in the CLI. Use the following command to create a dynamic ACL:

```
create access-list dynamic_rule conditions actions {non_permanent}
```

As an example of creating a dynamic ACL rule, compare an ACL policy file entry with the CLI command that creates the equivalent dynamic ACL rule.

The following ACL policy file entry will drop all *ICMP (Internet Control Message Protocol)* echo-requests:

```
entry  icmp-echo {
    if  {
        protocol  icmp;
        icmp-type  echo-request;
    } then {
        deny;
    }
}
```

To create the equivalent dynamic ACL rule, use the following command:

```
create access-list icmp-echo "protocol icmp;icmp-type echo-request"
"deny"
```

Notice that the *conditions* parameter is a quoted string that corresponds to the match conditions in the if { ... } portion of the ACL policy file entry. The individual match conditions are concatenated into a single string. The *actions* parameter corresponds to the then { ... } portion of the ACL policy file entry.

From the command line you can get a list of match conditions and actions by using the following command:

```
check policy attribute {attr}
```

The ACL rule shown in the example will be saved when the save command is executed, because the optional keyword non-permanent was not configured. This allows the rule to persist across system reboots.

Note also that the sample ACL rule does not specify an application to which the rule belongs. The default application is CLI.

Limitations

Dynamic ACL rule names must be unique, but can be the same as used in a policy file-based ACL. Any dynamic rule counter names must be unique. CLEAR-FLow rules can be specified only in policy files and therefore apply only to rules created in a policy file.

IPv6 destination address cannot be combined with L4 source port (or range) or ethernet type in the same DACL for role-based users.

Configuring the ACL Rule on the Interface

After a dynamic *ACL* rule has been created, it can be applied to a port, *VLAN*, or to the wildcard any interface. When the ACL is applied, you specify the precedence of the rule among the dynamic ACL rules. To configure the dynamic ACL rule on an interface, use the following command:

```
configure access-list add dynamic_rule [ [[first | last] {priority
p_number} {zone zone} ] | [[before | after] rule] | [ priority p_number
{zone zone} ]] [ any | vlan vlan_name | ports port_list ] {ingress |
egress}
```

To remove a dynamic ACL from an interface, use the following command:

```
configure access-list delete ruleName [ any | vlan vlan_name | ports
port_list | all] {ingress | egress}
```

An ACL can be created to be used when an edge port detects a loop. This ACL acts to block looped frames while allowing the port to remain in a forwarding state rather than shutting down. To configure a dynamic ACL for blocking looped *STP* BPDUs on port 6, for example, use the following:

```
create access-list bpdu1 "ethernet-destination-address    \
                          01:80:C2:00:00:00;" "deny; count bpdu1"
conf access-list add "bpdu1" first ports 6 ingress
```

To configure a dynamic ACL for blocking PVST frames on port 6, use the following:

```
create access-list bpdu2 "ethernet-destination-address \
                                01:00:0c:cc:cc:cd;" "deny; count bpdu2"
 conf access-list add "bpdu2" first ports 6 ingress
```

To unconfigure the STP ACL, use the following:

```
conf access-list del "bpdu1" ports 6
del access-list "bpdu1"
```

Configuring ACLs on a Management Port

Hardware *ACL* support is not possible on the management port. Untagged packets that are received on the management port are processed in software and can be filtered using ACLs. ACLs applied to the management port/vlan are installed only in software and not in the hardware.

For example, to block an *ICMP* echo-request on a management port, use the following:

```
create access-list echo "protocol icmp; icmp-type echo-request;" "deny; count echo"
conf access-list add "echo" first vlan "Mgmt" ingress
```

To unblock ICMP echo request on a management port, use the following:

```
conf access-list del "echo" vlan "Mgmt"
del access-list "echo"
```

To show ACL dropped packet counters, use the following command:

```
show access-list dynamic counter
```

Configuring ACL Priority

Management of *ACLs* is flexible, with configurable priority for dynamic ACLs. This includes ACLs inserted by internal and external applications, as well as those inserted using the CLI. The priority is assigned by a system of zones, and within zones by numeric codes.

Zones are of two types:

- System Space—The System Space zones include the following:
  - SYSTEM_HIGH—This zone always has the highest priority.
  - SYSTEM_LOW—This zone always has the lowest priority.

  The priorities cannot be changed.

  No configuration is allowed by the user into System Space.

  Hal is the only application in a System Space zone.
- User Space—The User Space zones include the following:
  - DOS—This is the denial of service zone.
  - SYSTEM—This is the zone for applications that require a CPU-copy or mirror and for redirect ACLs.
  - SECURITY—This is the zone for ACLs installed by security appliances and internal security processes.

  User Space zones consist of default zones and created zones. Default zones group like functions and cannot be deleted.

  The administrator has the ability to create new zones and configure the priority of both default and created zones. See Configuring User Zones on page 374 for discussion of created zones and applications. Applications insert ACLs into zones.

To view both System Space and User Space zones, use the `show access-list zone` command.

Table 27 shows the priority of System Space zones and User Space zones together with the default assignments and priority of applications by zone.

Table 27: Default Assignment and Priority of Applications, by Zone

Zone/Default Application	Default Priority	Platform
SYSTEM SPACE ZONES		
hal	1	
USER SPACE ZONES		
DOS	2	
hal	1	All platforms
Dos	2	All platforms
SYSTEM	3	

Table 27: Default Assignment and Priority of Applications, by Zone (continued)

Zone/Default Application	Default Priority	Platform
Cli	1	All platforms
IpSecurity	2	All platforms
NetLogin	6	All platforms
SECURITY	4	
GenericXml (Allows configuration of one additional external application)	4	All platforms
SYSTEM SPACE ZONES		
hal	1	

Note
The priority of static ACLs is determined by the order they are configured, with the first rule configured having the highest priority.

Configuring User Zones

There is a configurable process for applications to insert an *ACL* into a zone according to the priority of the application within that zone. Applications can occupy multiple zones. For example, you can add the Cli application to the DOS zone, and assign it a higher priority than the Dos application. The DOS zone then has two applications, Cli and Dos application, and within the DOS zone, an ACL created by the Cli has a higher priority than an ACL inserted by the Dos application.

Another way to configure ACL priority is by creating new zones. For example, you might create a zone called MY_HIGH_ZONE, and assign that zone a priority below the DOS zone and above the System zone. You can add applications to that zone and assign their priority.

The example below shows the ACL zone priority that would result from adding the MacInMac and Cli applications to MY_HIGH_ZONE:

1. SYSTEM_HIGH_ZONE

   hal
2. DOS Zone

   hal

   DoS
3. MY_HIGH_ZONE

   MacInMac

   Cli

4. SYSTEM Zone
   Dot1Ag
   Dot1AgDefault
   MacInMac
   Cli
   NetLogin
5. SECURITY Zone
   FlowVSR
   FlowVSRTS
   Generic Xml
6. SYSTEM_LOW_ZONE
   hal

Applications can insert an ACL into any of the zones to which the application belongs.

If an application attempts to insert an ACL into a zone where the application is not configured, an error message appears, and the ACL is not installed. Therefore, you have full control of ACL priorities and you can configure the switch to install ACLs from an application at any priority level. In the example above, the application Cli can insert an ACL into either MY_HIGH_ZONE or the SYSTEM zone. The location of the ACL within the zone depends on the priority assigned by the application. An application can assign priority to an ACL using:

- priority attributes (first or last)
- relative priority
- priority numbers

The priority attributes first (highest priority) and last (lowest priority) can be applied to an ACL to establish its position within a zone.

Relative priority sets the ACL priority relative to another ACL already installed by the application in the same zone.

Priority numbers allow an application to specify the priority of an ACL within a zone. The priority numbers are unsigned integers from 0 to 7; a lower number represents a higher priority. This means that if an application adds an ACL at priority 5 and later adds another ACL at priority 3, the second ACL has higher priority.

If an application assigns the same priority number to two ACLs, the ACL added most recently has the higher priority. It is inserted in the priority map immediately ahead of the older ACL that has the same priority number. This effectively allows the application to create sub-zones within a zone. The attributes first and last can be used in combination with priority numbers to prioritize the ACLs within a sub-zone. For example, an ACL could be configured with the first attribute, along with the same priority number as other ACLs in the same zone, effectively assigning that ACL the highest priority within a sub-zone.

The `show configuration` command shows the current configuration of the entire switch in the form of CLI commands which can later be played back to configure the switch.

The `show configuration acl` command shows the current configuration of the ACL manager.

The new application keyword allows you to specify the application to which the ACL will be bound. Typically, applications create and insert ACLs on the switch; however the administrator can install ACLs "on behalf" of an application by specifying the application keyword. (This keyword is also used with the show config acl command to enable CLI playback). If no application is specified, the default application is CLI.

This means you have the ability to create, delete, and configure ACLs for any application.

- To create a zone, use the following command:

  `create access-list zone` *name* **zone-priority***number*

- To configure the priority of zones, use the following command:

  `configure access-list zone` *name* **zone-priority** *number*

- To add an application to a zone at a particular priority, or to change the priority of an application within a zone, use the following command:

  `configure access-list zone` *name* {**add**} **application***appl-name* **application_priority** *number*

  An application must occupy at least one zone.

- To move an application within a zone or to another zone, use the following command:

  `configure access-list zone` *name* **move-application***appl-name* `to-zone`*number* **application-priority***number*

  All applications can be configured to go into any and all zones.

  A change in the zone list results in a change in the order of dynamic ACLs that have been applied per interface. The changes in hardware are achieved by uninstalling and then reinstalling the dynamic ACLs in the new positions. There is a possibility, due to hardware constraints, that some ACLs will not be reinstalled. These occurrences are logged.

- To delete an application from a zone, use the following command:

  `configure access-list zone` *name* **delete application** *appl-name*

  When you delete an application from a zone, any ACLs that have been inserted into that zone for the deleted application are moved to the next higher zone in which the application appears.

- To delete a zone, use the following command:

  `delete access-list zone` *name*

You must remove all applications from a zone before you can delete the zone. You cannot delete the default zones.

Network-Zone Support in ACLs

ExtremeXOS Network-Zone support allows you to create a network-zone, add multiple IP addresses and/or MAC addresses to it, and use the network-zone in policy files.

This feature provides the ability to add a single attribute “source-zone,” or “destination-zone” to an entry of a policy file. This entry is then expanded into multiple entries depending upon the number of IP and/or MAC addresses configured in that particular zone. If the zone is added to the policy with the keyword “source-zone,” the attributes that are configured in that particular zone are added as either a “source-address” or an “ethernet-source-address” in the policy. Conversely, if the network-zone is added as a “destination-zone,” the attributes are added to the policies as a “destination-address,” or an “ethernet-destination-address.”

After you make changes in the zones and issue a refresh of a specific zone, or all the zones, the policies that have the corresponding zones configured as source-zone or destination-zone in their entries are expanded and refreshed in the hardware.

If you configure the following policy to a port or *VLAN*, or through applications like IdMgr or *Extreme Network Virtualization (XNV)*,

```
Policy: test
entry e1 {
if match all {
source-zone zone1 ;
}
Then {
permit  ;
}
}
```

and the network-zone “zone1” that is part of the policy is configured as below:

```
create access-list network-zone zone1
configure access-list network-zone zone1 add ipaddress 10.1.1.1 255.255.255.255
configure access-list network-zone zone1 add ipaddress 10.1.1.1 255.255.255.240
configure access-list network-zone zone1 add ipaddress 12.1.1.0 255.255.255.0
```

When you refresh the network-zone “zone1,” the policy is expanded as follows, and is applied in the hardware:

```
entry Cl0:0_10.1.1.1_E1 {
if match all {
source-address 10.1.1.1 / 32 ;
} then {
permit  ;
}
}
entry Cl0:0_10.1.1.1_E2 {
if match all {
source-address 10.1.1.1 / 28 ;
} then {
```

```
permit  ;
}
}

entry Cl0:0_12.1.1.0_E3 {
if match all {
source-address 12.1.1.0 / 24 ;
} then {
permit  ;
}
}
```

When the policy is configured with the network-zone, the zone may or may not have attributes configured in it. In cases where the network-zone does not have any attributes, the policy is created with entries that do not have the network-zone attribute alone.

So, if you create the following policy:

```
Policy: test2
entry e1 {
if match all {
source-zone zone2 ;
protocol udp ;
}
Then {
permit  ;
}
}
entry e2 {
if match all {
protocol tcp ;
}
Then {
permit  ;
}
}
```

And the network-zone “zone2” is just created, but is not configured with any attributes, the policy appears as follows and has only the second entry “e2,” and not “e1”.

```
entry e2 {
protocol tcp;
}
Then {
permit;
}
}
```

Once the network-zone “zone2” is configured with one or more attributes, and refreshed, the policy is updated accordingly. In this instance, the name of the entries that have a source-zone or a destination-zone are changed. This is because each entry in the original policy that has a source-zone/destination-zone is converted to a maximum of eight entries in the new policy.

A single policy can have one or more network-zones configured in it. It can also have the same network-zone in multiple entries with different combinations, as well as support for other attributes in the policy file. Similarly, the same network-zone can be

configured to multiple policies. In cases where the policy has multiple network-zones, and only some of those network-zones are refreshed, the entries that correspond to those specific network-zones are alone refreshed, and not entries that correspond to the other network-zones.

After you refresh a network-zone, all the policies that have the specified network-zone are modified, and a refresh for each of those policies is sent to the hardware. The command succeeds only after receiving a successful return for all the policies that have this particular network-zone. If for some reason one of the policy's refresh fails in the hardware, all the policies that are supposed to refresh are reverted back to their previous successful state, and the command is rejected.

Additionally, the configuration or refresh can fail if the attributes in the network-zone are not compatible with the attributes in the corresponding entries of the policy. For example, in platforms that do not support wide-key or UDF, a policy entry cannot have Layer 2 attributes and Layer4 attributes. In such cases, if the entry has "protocol tcp" and a network-zone that has an ethernet source address, the configuration fails in the hardware.

In cases where the refresh fails, the content of the policy and the content of the network-zone may go out of sync, because the policy reverts back to the last successful state, whereas the network-zone will contain the last configured values.

Here is an example:

```
create access-list network-zone zone1
configure access-list network-zone zone1 add ipaddress 10.1.1.1/32
configure access-list network-zone zone1 add ipaddress 10.1.1.1/28
```

Once this configuration is refreshed and is successfully installed in the hardware, the policy will look like the following:

```
entry Cl0:0_10.1.1.1_E1 {
if match all {
source-address 10.1.1.1 / 32 ;
} then {
permit  ;
}
}
entry Cl0:0_10.1.1.1_E2 {
if match all {
source-address 10.1.1.1 / 28 ;
} then {
permit  ;
}
}
```

If you remove 10.1.1.1/28, and adds 10.1.1.1/24 to the network-zone and perform a refresh,

```
configure access-list network-zone zone1 delete ipaddress 10.1.1.1/28
configure access-list network-zone zone1 add ipaddress 12.1.1.0/24
```

and if for some reason the policy refresh fails, the policy and the network-zone will look like this:

```
entry Cl0:0_10.1.1.1_E1 {
if match all {
source-address 10.1.1.1 / 32 ;
} then {
permit  ;
}
}
entry Cl0:0_10.1.1.1_E2 {
if match all {
source-address 10.1.1.1 / 28 ;
} then {
permit  ;
}
}

create access-list network-zone zone1
configure access-list network-zone zone1 add ipaddress 10.1.1.1 255.255.255.255
configure access-list network-zone zone1 add ipaddress 12.1.1.0 255.255.255.0
```

Configuring Network-Zone Support in ACLs

Use the following command to configure the network-zone support in ACLs.:

Creating a Network-Zone

To create a network-zone with the specified name, enter the command below. You can then associate this network-zone with the policy file using either the *source-zone* or *destination-zone* attribute.

`create access-list network-zone` *`zone_name`*

Here is an example:

```
Switch# create access-list network-zone zone1
```

If you try to create a network-zone that is already created, the following error message is displayed on the console, and the command is rejected:

```
Switch# create access-list network-zone zone1
Error:  Network Zone "zone1" already exists.
```

Deleting a Network-Zone

To delete a network-zone, and all the configurations that belong to the zone, enter the following command:

`delete access-list network-zone` *`zone_name`*

Here is an example:

```
Switch# delete access-list network-zone zone1
```

If you try to delete a network-zone that is bound with one or more policy files, the following error message is displayed, and the command is rejected:

```
Switch# delete access-list network-zone zone1
Error:  Network Zone "zone1" - Unable to delete zone. Zone has one
or more policies.
```

Adding or Removing Network-Zone Attributes

To add or remove IP/MAC addresses to or from the network-zone, enter the following command:

```
configure access-list network-zone zone_name [add | delete] [mac-address macaddress {macmask} | ipaddress [ipaddress {netmask} | ipNetmask | ipv6_address_mask]]
```

Here is an example:

```
Switch# configure access-list network-zone zone1 add ipaddress 11.1.1.1/24
```

If you try to add the same IP/MAC with the same or narrow mask, the configuration is rejected, with the following error message:

```
Switch# configure access-list network-zone "zone1" add ipaddress 11.1.1.1/32
Error:  Network Zone "zone1" - Zone already has the same entity value with same or wider mask.
```

If the you try to add more than eight attributes to a network-zone, the following error message is issued:

```
Switch# configure access-list network-zone "zone1" add ipaddress 11.1.1.1/24
Error:  Network Zone "zone1" - Reached maximum number of attributes. Unable to add more.
```

Refreshing Network-Zones

To refresh a specific network-zone, or all the network-zones, enter the following command:

```
refresh access-list network-zone [zone_name | all]
```

Here is an example:

```
Switch# refresh access-list network-zone zone1
```

Note

When you issue the command to refresh a network-zone, or all network-zones, it can take a long time to clear the CLI because each individual policy must be converted before it is refreshed. The command succeeds, or fails, only after it receives a success response for all policy refresh, or when a first refresh failure is received from the hardware.

If the refresh fails for a specific zone, the following error message is printed on the console.

```
Switch# refresh access-list network-zone zone1
Error: Refresh failed for network-zone "zone1".
```

Monitoring Network-Zone Support in ACLs

show access-list network-zone

To monitor various network statistics, use the `show access-list network-zone` command. This command's output displays the network-zones configured, the number of attributes configured, and the number of policy files that has the specific zones in it.

Example

```
Switch# sh access-list network-zone
============================================================
Network Zone                        No. of    No. of Policies
Entities       Bound
============================================================
zone1                                 5              2
zone2                                 3              1
zone3                                 0              0
============================================================
Total Network Zones : 3
```

The following example displays detailed information about a specific network zone, the attributes configured in the zone, and the policies bound to the zone:

```
Switch# show access-list network-zone zone1
Network-zone     : zone1
Total Attributes : 3
Attributes       : 10.1.1.1 / 32
10.1.1.1 / 30
10.1.1.0 / 24
No. of Policies  : 1
Policies         : test
Switch # sh access-list network-zone zone2
Network-zone     : zone2
No. of Entities  : 3
Entities         : 00:00:00:00:00:22 / ff:ff:ff:ff:ff:ff
00:00:00:00:00:23 / ff:ff:ff:ff:00:00
00:00:00:00:00:24 / ff:ff:ff:ff:ff:00
No. of Policies : 0
```

CCOS ACL Match Criteria

This feature adds support for the ExtremeXOS *ACL* match criteria `ccos`. It provides the ability to specify access-lists that filter on the inner-dot1p field of double tagged packet.

```
entry one {

if {

ccos 3;
} then {

meter m1;
count c1;
}
}
```

ACL Evaluation Precedence

The section Precedence on page 383 describes *ACL* evaluation precedence for different platforms.

Precedence

This section describes the precedence for evaluation among *ACL* rules for ExtremeSwitching series switches. In many cases there will be more than one ACL rule entry for an interface. This section describes how multiple rule entries are evaluated.

Multiple rule entries do consume hardware resources. If you find that your situation runs up against those limits, there are steps you can take to conserve resources by modifying the order of the ACL entries that you create. For details, see ACL Mechanisms on page 388.

Rule Evaluation

When there are multiple rule entries applied to an interface, evaluation proceeds as follows:

- A packet is compared to all the rule entry match conditions at the same time.
- For each rule where the packet matches all the match conditions, the action and any action modifiers in the then statement are taken. If there are any actions or action modifiers that conflict (deny vs. permit, etc), only the one with higher precedence is taken.
- If a packet matches no rule entries in the *ACL*, it is permitted.

Often there will be a lowest-precedence rule entry that matches all packets. This entry will match any packets not otherwise processed, so that the user can specify an action to overwrite the default permit action. This lowest-precedence rule entry is usually the last entry in the ACL policy file applied to the interface.

Precedence of Dynamic ACLs

Dynamic ACLs have a higher precedence than any ACLs applied using policy files. The precedence among any dynamic ACLs is determined as they are configured. The precedence of ACLs applied using policy files is determined by the rule's relative order in the policy file.

Precedence of L2/L3/L4 ACL Entries

Rule precedence is solely determined by the rule's relative order. L2, L3, and L4 rules are evaluated in the order found in the file or by dynamic *ACL* configuration.

Precedence Among Interface Types

As an example of precedence among interface types, suppose a physical port 1:2 is a member port of the *VLAN* yellow. *ACL*s could be configured on the port, either singly or as part of a port list, on the VLAN yellow, and on all ports in the switch (the wildcard ACL). For all packets crossing this port, the port-based ACL has highest precedence, followed by the VLAN-based ACL and then the wildcard ACL.

Precedence with Egress ACLs

Egress *ACL*s are supported on all ExtremeSwitching Universal switches. For these, egress ACL lookup happens at egress, and diffserv, dot1p and other non-ACL feature examination happen at ingress. Therefore, egress ACL happens at the last moment and has precedence.

Redundant Rules

For ExtremeSwitching series switches, eliminate redundant rules (any with the EXACT same match criteria) in the policy file. If two rules have identical match conditions, but different actions, the second rule is rejected by the hardware.

For example, the two following *ACL* entries are not allowed:

```
entry DenyNMR {
    if {
        protocol 17;
        destination-port 161;
    } then {
        deny;
        count denyNMR;
    }
}
entry DenyNIC {
    if {
        protocol 17;
        destination-port 161;
    } then {
        deny;
        count denyNIC;
    }
}
```

Applying ACL Policy Files

A policy file intended to be used as an *ACL* is applied to a port, *VLAN*, or to all interfaces (the any keyword). Use the name of the policy file for the *aclname* parameter in the CLI command. To apply an ACL policy, use the following command:

`configure access-list` *aclname* [**any** | **ports** *portlist* | **vlan** *vlanname*] {**ingress** | **egress**}

If you use the any keyword, the ACL is applied to all the interfaces and is referred to as the wildcard ACL. This ACL is evaluated for any ports without specific ACLs, and it is also applied to any packets that do not match the specific ACLs applied to the interfaces.

If an ACL is already configured on an interface, the command will be rejected and an error message displayed.

To remove an ACL from an interface, use the following command:

`unconfigure access-list` *aclname* [**any** | **ports** *portlist* | **vlan** *vlanname*] {**ingress** | **egress**}

To display which interfaces have ACLs configured, and which ACL is on which interface, use the following command:

```
show access-list aclname [any | ports portlist | vlan vlanname] {ingress
| egress}
```

Note
If an ACL needs to be installed for traffic that is L3 routed, and the ingress/egress ports are on different packet-processing units or different slots, and any of the following features are enabled, we recommend that you install the policy on a per-port basis rather than applying it as a wildcard, or VLAN-based ACL.

-
- PVLAN
- Multiport-*FDB (forwarding database)*

Displaying and Clearing ACL Counters

To display the *ACL* counters, use the following command:

```
show access-list counter {countername} {any | ports port_list | vlan
vlan_name} {ingress | egress}
```

To clear the access list counters, use the following command:

```
clear access-list {dynamic} counter {countername} {any | ports port_list
| vlan vlan_name} {ingress | egress}
```

Example ACL Rule Entries

The following entry accepts all the UDP packets from the 10.203.134.0/24 subnet that are destined for the host 140.158.18.16, with source port 190 and a destination port in the range of 1200 to 1250:

```
entry  udpacl {
    if  {
        source-address 10.203.134.0/24;
        destination-address 140.158.18.16/32;
        protocol  udp;
        source-port 190;
        destination-port  1200 - 1250;
    } then {
        permit;
    }
}
```

The following rule entry accepts TCP packets from the 10.203.134.0/24 subnet with a source port larger than 190 and ACK & SYN bits set and also increments the counter tcpcnt. The packets will be forwarded using *QoS* profile QP3.

Note
These operators <,<=,> and >= are supported only in policy ACL mode and not in Dynamic ACL mode.

```
entry  tcpacl {
    if  {
```

```
        source-address 10.203.134.0/24;
        protocol  TCP;
        source-port > 190;
        tcp-flags  syn_ack;
    } then {
        permit;
        count tcpcnt ;
        qosprofile qp3;
    }
}
```

The following example denies *ICMP* echo request (ping) packets originating from the 10.203.134.0/24 subnet, and increments the counter icmpcnt:

```
entry  icmp {
    if  {
        source-address 10.203.134.0/24;
        protocol  icmp;
        icmp-type  echo-request;
    } then {
        deny;
        count icmpcnt;
    }
}
```

The following example prevents TCP connections from being established from the 10.10.20.0/24 subnet, but allows established connections to continue, and allows TCP connections to be established to that subnet. A TCP connection is established by sending a TCP packet with the SYN flag set, so this example blocks TCP SYN packets.

```
entry  permit-established {
    if  {
        source-address 10.10.20.0/24;
        protocol  TCP;
        tcp-flags  syn;
    } then {
        deny;
    }
}
```

The following entry denies every packet and increments the counter default:

```
entry  default {
    if  {
    } then {
        deny;
        count default;
    }
}
```

The following entry permits only those packets with destination MAC addresses whose first 32 bits match 00:01:02:03:

```
entry rule1 {
    if {
        ethernet-destination-address 00:01:02:03:01:01 ff:ff:ff:ff:00:00 ;
    } then {
        permit  ;
```

```
    }
}
```

The following entry denies IPv6 packets from source addresses in the 2001:db8:c0a8::/48 subnets and to destination addresses in the 2001:db8:c0a0:1234::/64 subnets:

```
entry ipv6entry {
    if {
        source-address 2001:DB8:C0A8:: / 48;
        destination-address 2001:DB8:C0A0:1234:: / 64;
    } then {
        deny;
    }
}
```

Access lists have entries to match an Ethernet type, so be careful when configuring access lists to deny all traffic. For example, the following rule entries permit traffic only to destination 10.200.250.2 and block any other packet.

```
entry test_policy_4 {
    if {
        source-address 0.0.0.0/0;
        destination-address 10.200.250.2/32;
    } then {
        permit;
        count test_policy_permit;
    }
}
# deny everyone else
entry test_policy_99 {
    if {
    } then {
        deny;
        count test_policy_deny;
    }
}
```

Since the deny section does not specify an Ethernet type, all traffic other than IP packets destined to 10.200.250.2/32 are blocked, including the ARP packets. To allow ARP packets, add an entry for the Ethernet type, 1x0806, as shown below.

```
entry test_policy_5 {
    if {
        ethernet-type 0x0806;
    } then {
        permit;
        count test_policy_permit;
    }
}
```

The following entries use vlan-ids to set up meters based on individual VLANs.

```
myServices.pol
    entry voiceService {
        if {
            vlan-id 100;
```

```
        } then {
            meter voiceServiceMeter;
        }
    }
    entry videoService {
        if {
            vlan-id 101;
        } then {
            meter videoServiceMeter;
        }
    }
…and so on.
```

To bind this ACL to a port with vlan-id match criteria use the following command:

```
config access-list myServices port <N>
```

The following entry shows how to take action based on *VLAN* tag priority information. In this example, the dot1p match keyword is used to allow and count every tagged packet with a VLAN priority tag of 3.

```
entry count_specific_packets {
    if {
        dot1p 3;
    } then {
        count allowed_pkts;
        permit;
    }
}
```

ACL Mechanisms

For many applications of *ACL*s, it is not necessary to know the details of how ACLs work. However, if you find that your application of ACLs is constrained by hardware limitations, you can often rearrange the ACLs to use the hardware more efficiently. The following sections go into some detail about how the ACLs use hardware resources, and provide some suggestions about how to reorder or rewrite ACLs to use fewer resources.

- ACL Slices and Rules.
- ACL Counters—Shared and Dedicated.

ACL Slices and Rules

ExtremeSwitching switches use a mechanism different from the earlier series to implement *ACL*s. The same architecture and guidelines apply to both platforms.

Instead of the per port masks used in earlier switches, these platforms use slices that can apply to any of the supported ports. An ACL applied to a port may be supported by any of the slices.

The slice support is as follows:

- 4120 switches—
  - Eight ingress stage slices with 2,000 single-wide rules
  - Four VLAN look-up stage slices with 512 rules

    - Four egress stage slices with 512 rules
- 4220 switches—
    - Eight ingress stage slices with 2,000 single-wide rules
    - Four VLAN look-up stage slices with 512 rules
    - Four egress stage slices with 512 rules
- 5120 switches—
    - Eight ingress stage slices with 1,000 single-wide rules
    - Eight ingress stage slices with 512 double-wide rules
    - Four egress stage slices with 256 rules
- 5320 switches—
    - Four VLAN look-up stage slices with 1,024 rules for 5320-48TP models.
    - Eighteen ingress stage slices with 16,000 single-wide rules for 5320-48T/P and 24 and 16-port models.
    - Eighteen ingress stage slices with 8,000 double-wide rules for 5320-48T/P and 24 and 16-port models.
    - Four egress stage slices with 1,024 rules for 5320-48T/P models.
    - Four VLAN look-up stage slices with 512 rules for 5320-24 and 16-port variants models.
    - Four egress stage slices with 512 rules for 5320-24 and 16-port variants.
- 5420 switches—
    - Four VLAN look-up stage slices with 1,024 rules.
    - Eighteen ingress stage slices with 36,000 single-wide rules for 5420M models.
    - Eighteen ingress stage slices with 16,000 single-wide rules for 5420F models.
    - Eighteen ingress stage slices with 18,000 double-wide rules for 5420M models.
    - Eighteen ingress stage slices with 8,000 double-wide rules for 5420F models.
    - Four egress stage slices with 1,024 rules.
- 5520 switches—
    - Four VLAN look-up stage slices with 2,048 rules.
    - Eighteen ingress stage slices with 18,000 single-wide rules.
    - Four egress stage slices with 1,024 rules.
- 5720 switches—
    - Four VLAN look-up stage slices with 2,048 rules.
    - Eighteen ingress stage slices with 36,000 single-wide rules for 5720-MXW models.
    - Twelve ingress stage slices with 18,000 single-wide rules for 5720-MW models.
    - Eighteen ingress stage slices with 18,000 double-wide rules for 5720-MXW models.
    - Twelve ingress stage slices with 9,000 double-wide rules for 5720-MW models.
    - Four egress stage slices with 2,048 rules.
- 7520 switches—
    - Four VLAN look-up stage slices with 1,024 rules.
    - Eighteen ingress stage slices with 36,000 single-wide rules.
    - Twelve ingress stage slices with 18,000 single-wide rules.

    - Eighteen ingress stage slices with 18,000 double-wide rules.
    - Twelve ingress stage slices with 9,000 double-wide rules.
    - Four egress stage slices with 2,048 rules.
- 7720 switches—
    - Four VLAN look-up stage slices with 1,024 rules.
    - Eighteen ingress stage slices with 36,000 single-wide rules.
    - Twelve ingress stage slices with 18,000 single-wide rules.
    - Eighteen ingress stage slices with 18,000 double-wide rules.
    - Twelve ingress stage slices with 9,000 double-wide rules.
    - Four egress stage slices with 2,048 rules.

ACL Allocation to Slices—All Platforms

A single slice can implement ACLs that are applied to more than one port. When an ACL entry is applied, if its match conditions do not conflict with an already existing ACL, the entry is added to the rule memory of an already populated slice.

When ACLs are applied, the system programs each slice to select parts of the packet information to be loaded into it. For example, one possible way a slice can be programmed allows it to hold the information about a packet's ingress port, source and destination IP address, IP protocol, source and destination Layer 4 ports, DSCP value, TCP flag, and if it is first fragments. Any rule entry that consists of match conditions drawn from that list is compatible with that slice. This list of conditions is just one example. A complete description of possible ways to program a slice is discussed in Compatible and Conflicting Rules on page 392.

In the following example, the two rule entries are compatible and require only one slice in hardware even though they are applied to different ports. The following entry is applied to port 1:

```
entry ex_A {
    if {
        source-address 10.10.10.0/24 ;
        destination-port 23 ;
        protocol tcp ;
    } then {
        deny ;
    }
}
```

and the following entry is applied to port 2:

```
entry ex_B {
    if {
        destination-address 192.168.0.0/16 ;
        source-port 1000 ;
        protocol tcp ;
    } then {
        deny ;
    }
}
```

Both of these ACLs could be supported on the same slice, since the match conditions are taken from the example list discussed earlier. This example is shown in the following figure. In the example, we refer to slice A, even though the slices are numbered. Slice A just means that one slice is used, but does not specify a particular slice. Some rules require more than one slice, so we use letters to show that different slices are used, but not which specific slices.

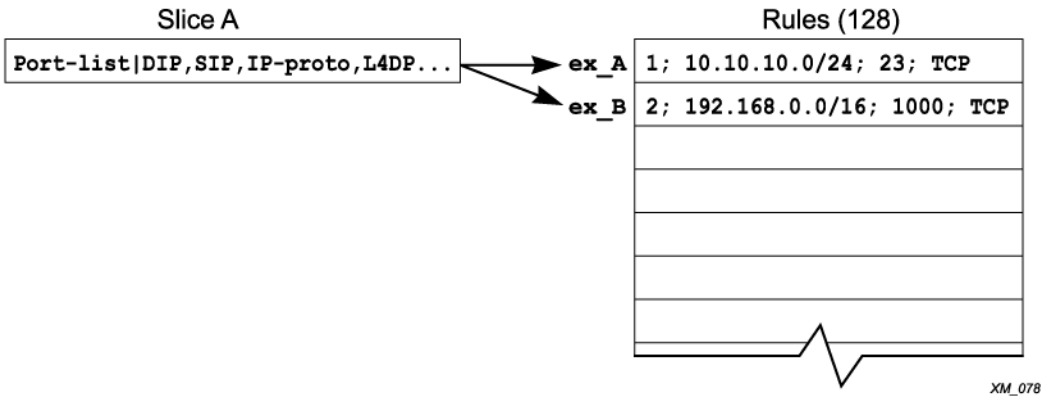


Figure 25: ACL Entry ex_A and ex_B

There are cases where compatible ACLs require using a different slice. If the memory associated with a slice is filled with rule entries, then another slice will be used to process any other compatible entries.

For example, consider the following 129 rule entries applied to ports 3-7:

```
entry one {
    if {
        source-address 10.66.10.0/24 ;
        destination-port 23 ;
        protocol tcp ;
    } then {
        deny ;
    }
}
entry two {
    if {
        destination-address 192.168.0.0/16 ;
        source-port 1000 ;
        protocol tcp ;
    } then {
        deny ;
    }
}
entry three {
    if {
        source-address 10.5.2.246/32 ;
        destination-address 10.0.1.16/32 ;
        protocol udp ;
        source-port 100 ;
        destination-port 200 ;
    } then {
        deny ;
    }
}
....
[The 125 intervening entries are not displayed in this example]
```

```
....
entry onehundred_twentynine {
    if {
        protocol udp ;
        destination-port 1714 ;
    } then {
        deny ;
    }
}
```

The following figure shows the result of applying the 129 entries; 128 of the entries are applied to one slice, and the final entry is applied to a different slice. If another compatible entry is applied from another port, for example, it will use Slice B.

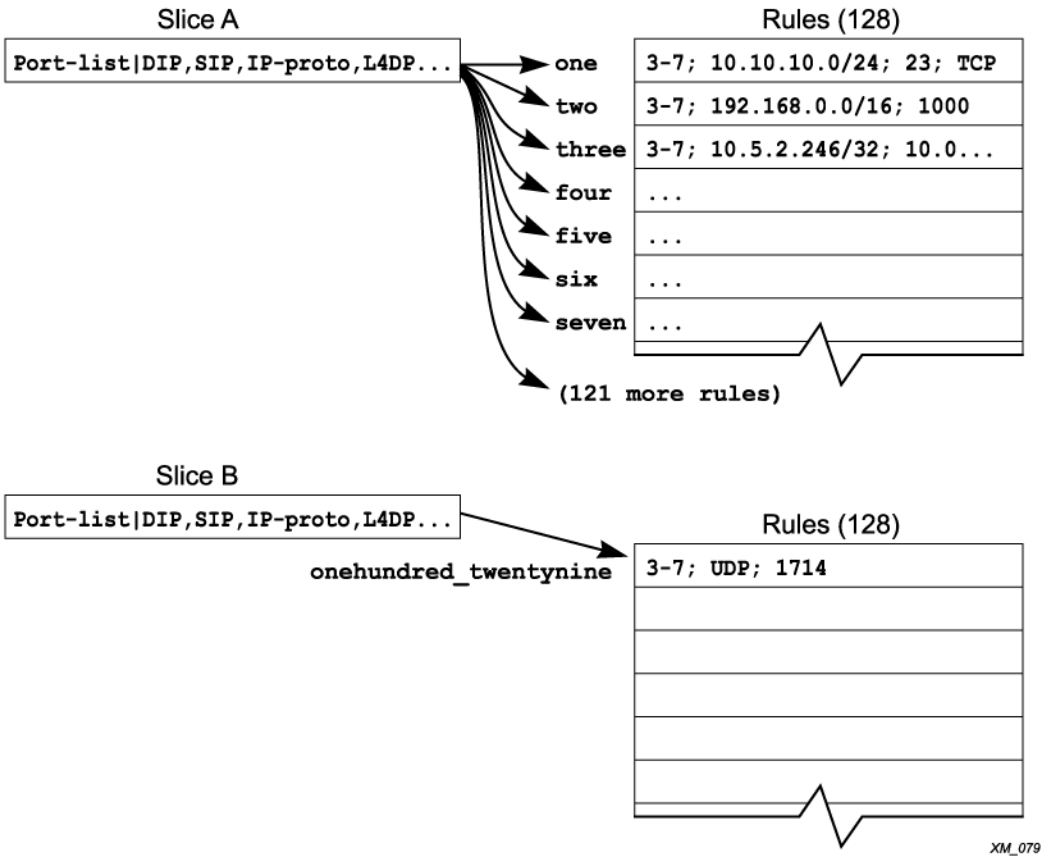


Figure 26: ACL Entry one Through onehundred_twentynine

As entries are configured on the switch, the slices are programmed to implement the rules, and the rule memory is filled with the matching values for the rules. If a compatible slice is available, each entry is added to that slice.

Compatible and Conflicting Rules

The slices can support a variety of different *ACL* match conditions, but there are some limitations on how you combine the match conditions in a single slice. A slice is divided up into fields, and each field uses a single selector. A selector is a combination of

match conditions or packet conditions that are used together. To show all the possible combinations, the conditions in Table 28 are abbreviated.

Table 28: Abbreviations Used in Field Selector Table

Abbreviation	Condition
Ingress	
DIP	destination address <prefix> (IPv4 addresses only)
DIPv6/128	destination address <prefix> (IPv6 address with a prefix length longer than 64)
DIPv6/64	destination address <prefix> (IPv6 address with a prefix length up to 64)
DSCP	dscp <number>
Etype	ethernet-type <number>
First Fragments	first ip fragments
FL	IPv6 Flow Label
Fragments	fragments
IP-Proto	protocol <number>
L4DP	destination-port <number> (a single port)
L4-Range	A Layer 4 port range. For example, if you specify “protocol UDP” and “port 200 - 1200” in an entry, you have used a Layer 4 range. There are a total of sixteen Layer 4 port ranges. Also, you can have a source port range, or a destination port range, but not both kinds of ranges together in the same entry.
L4SP	source-port <number> (a single port)
MACDA	ethernet-destination-address <mac-address> <mask>
MACSA	ethernet-source-address <mac-address>
NH	IPv6 Next Header field. Use protocol <number> to match. See IP-Proto
OVID	This is not a match condition used in ACLs, but is used when an ACL is applied to VLANs. An ACL applied to a port uses a different field selector than an ACL applied to a VLAN. VLAN IDs are outer VLAN IDs unless specified as inner VLAN IDs.
packet-type	This selector is used internally and not accessible by users through explicit ACLs.
Port-list	This is not a match condition used in ACLs, but is used when an ACL is applied to ports, or to all ports (the wildcard ACL). An ACL applied to a port uses a different field selector than an ACL applied to a VLAN.
SIP	source address <prefix> (IPv4 addresses only)
SIPv6/128	source address <prefix> (IPv6 address with a prefix length longer than 64)
SIPv6/64	source address <prefix> (IPv6 address with a prefix length up to 64)

Table 28: Abbreviations Used in Field Selector Table (continued)

Abbreviation	Condition
TC	IPv6 Traffic Class field. Use dscp <number>
TCP-Flags	TCP-flags <bitfield>
TPID	802.1Q Tag Protocol Identifier
TTL	Time-to-live
UDF	User-defined field. This selector is used internally and not accessible by users through explicit ACLs.
VID-inner	Inner VLAN ID
VRF	virtual router and forwarding instance
Egress	
DestIPv6	destination-address <ipv6>
DIP	destination-address
Etype	ethernet-type
IP-Proto	protocol
L4DP	destination-port. Support only single L4 ports and not port ranges.
L4SP	source-port. Support only single L4 ports and not port ranges.
MACDA	ethernet-destination-address
MACSA	ethernet-source-address
NH	IPv6 Next Header field.
SIP	source-address
SIPv6	source-address <ipv6>
TC	IPv6 Traffic Class field.
Tcp-Flags	tcp-flags
TOS	ip-tos or diffserv-codepoint
VlanId	vlan-id

The following ingress conditions are not supported on egress:

- fragments
- first-fragments
- IGMP-msg-type
- ICMP-type
- ICMP-code

Table 29 lists all the combinations of match conditions that are available. Any number of match conditions in a single row for a particular field may be matched. For example if Field 1 has row 1 (Port-list) selected, Field 2 has row 8 (MACDA, MACSA, Etype,

OVID) selected, and Field 3 has row 7 (Dst-Port) selected, any combination of Port-list, MACDA, MACSA, Etype, OVID, and Dst-Port may be used as match conditions.

If an ACL requires the use of field selectors from two different rows, it must be implemented on two different slices.

Table 29: Field Selectors for ExtremeSwitching Series Switches

Fixed Field	Field 1	Field 2	Field 3
Port-list	OVID, VID-inner	DIP, SIP, IP-Proto, L4DP, L4SP, DSCP, IPFlag, TCP-Flag	OVID
	Etype, OVID	DIP, SIP, IP-Proto, L4DP, L4SP, DSCP, IpInfo(First-Fragments, Fragments) TCP-Flag	OVID, IpInfo(First-Fragments, Fragments)
	VID-inner	DIPv6/128	OVID, VID-inner
	IpInfo(First-Fragments, Fragments), OVID	SIPv6/128	OVID, Etype
	OVID	DIPv6/64, IP-Proto, DSCP, FL, TCP-Flag	VID-Inner
	IP-Proto, DSCP	MACDA, MACSA, OVID, Etype	L4-Range
	"User Defined Field" 1b	MACSA, OVID, Etype, SIP	FL
		MACDA, OVID, Etype, DIP, IP-Proto	UDF1[95..64]
		"User Defined Field" 1	
		"User Defined Field" 2	
		DIPv6/64, SIPv6/64	

Egress ACLs

Each of the four egress slices can be configured to one of the three combinations below. The rules that can be installed into a particular slice should be a subset of the combination to which that slice is configured.

The following are available combinations:

- Combination 1:

  ```
  <vlan-id, ethernet-source-address, ethernet-destination-address,
  ethernet-type>
  ```

- Combination 2:

  ```
  <vlan-id, diffserv-codepoint/ip-tos, destination-address, source-
  address, protocol, destination-port, source-port, tcp-flags>
  ```

- Combination 3:

  ```
  <vlan-id, ip-tos, destination-address<ipv6>, source-address<ipv6>,
  protocol>
  ```

Use the the tables inCompatible and Conflicting Rules on page 392 to determine which *ACL* entries are compatible. If the entries are compatible, they can be on the same slice.

For example, the earlier example entries are applied to ports:

```
entry ex_A {
    if {
        source-address 10.10.10.0/24 ;
        destination-port 23 ;
        protocol tcp ;
    } then {
        deny ;
    }
}
entry ex_B {
    if {
        destination-address 192.168.0.0/16 ;
        source-port 1000 ;
    } then {
        deny ;
    }
}
```

Entry ex_A consists of the following conditions (using the abbreviations from the following table), SIP, L4DP, and IP-Proto. Entry ex_B is DIP, L4SP. Since they are applied to ports, the selector for Field 1 is Port-list (the first item). The selector for Field 2 would be the first item, and Field 3 could be any item.

Our other example entries are also compatible with the entries ex_A and ex_B:

```
entry one {
    if {
        source-address 10.66.10.0/24 ;
        destination-port 23 ;
        protocol tcp ;
    } then {
        deny ;
    }
}
entry two {
    if {
        destination-address 192.168.0.0/16 ;
        source-port 1000 ;
    } then {
        deny ;
    }
}
entry three {
    if {
        source-address 10.5.2.246/32 ;
        destination-address 10.0.1.16/32 ;
        protocol udp ;
        source-port 100 ;
        destination-port 200 ;
    } then {
        deny ;
    }
}
```

Entry one is SIP, L4DP, and IP-Proto; entry two is DIP, and L4SP; entry three is SIP, DIP, IP-Proto, L4SP, and L4DP. All of these examples can use the first item in Field 2 in the tables.

However, if we add the following entry:

```
entry alpha {
    if {
        ethernet-destination-address 00:e0:2b:11:22:33 ;
    } then {
        deny ;
    }
}
```

This will not be compatible with the earlier one. Entry alpha is MACDA, and there is no MACDA in the first item for Field 2. Any entry with MACDA will have to use selector 7 or 8 from the following table (or 6 or 7 from the following table, depending on the platform). If an entry requires choosing a different selector from the table, it is not compatible and must go into a different slice.

Single Virtual Group for User ACLs

Prior to ExtremeXOS 16.1, when two user rules in two separate slices are matched by a packet, the non-conflicting actions from both of the rules are executed. This feature allows you to put all user rules into a single virtual group. When rules are in a single virtual group, even when two rules in two separate virtual slices are matched, only the actions of the highest precedence rule are executed. In effect, in this mode, multiple slices behave as a big single virtual slice.

Normally *ACL* hardware works in the following way: on arrival of a packet, all N slices are searched in parallel to find a possible match in each of these N slices. In each slice, upon finding the first match, the search within that slice stops. In other slices the search continues until a match is found or the end of the slice is reached. Thus, a single slice can produce only a single match, but all N slices combined together can produce up to N matches for a given packet.

In the case of multiple matches in multiple slices, all the actions of the rule in the highest priority virtual slice are executed. In addition, all the actions from the lower priority rules from the lower priority virtual slices are executed if those additional actions do not conflict with the actions of the highest priority rule. An example of non-conflicting with each other actions would be "permit" and "count". An example of conflicting with each other actions would be "permit" and "deny".

However, in more recent chipsets a new mode of operation was introduced where you can combine a few virtual slices into one big virtual group. In this mode of operation, even if a packet gets multiple matches from multiple virtual slices within the same virtual group, only the actions of the highest priority rule are executed, whereas the actions from the lower priority rules are not executed at all, even if those actions do not conflict with the actions of the highest priority rule.

This feature allows choosing between the old way of operation where every virtual slice is in its own virtual group and multiple matches are possible, and the new way, where

all user ACL's virtual slices are in the same virtual group and multiple matches are not possible.

Note
Some platforms that do not support virtual groups. On those chipsets even if single virtual group feature is enabled, ACL will operate in the old way and multiple matches would be still possible.

Rule Evaluation and Actions

When a packet ingresses the switch, its header is loaded into all the slices, and the header value is compared with the rule values. If the values match, the rule action is taken. Conflicting actions are resolved by the precedence of the entries. However, if rule entries are on different slices, then *ACL* counters can be incremented on each slice that contains a counter-incrementing rule.

Slice and Rule Use by Feature

A number of slices and rules are used by features present on the switch. You consume these resources when the feature is enabled.

- dot1p examination - enabled by default - 1 slice, 8 rules per chip
  - Slice A (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=packet-type)
- *IGMP (Internet Group Management Protocol)* snooping - enabled by default - 2 slice, 2 rules
  - Slice A (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=packet-type)
  - Slice B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=IP-Proto, TOS)
- *VLAN* without IP configured - 2 rules - 2 slices
  - Slice A (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=packet-type)
  - Slice C (F1=Port-list, F2=SIP, DIP, IP-proto, L4SP, L4DP, DSCP, F3=packet-type)
- IP interface - disabled by default - 2 slices, 3 rules (plus IGMP snooping rules above)
  - Slice A (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=packet-type)
  - Slice C (F1=Port-list, F2=SIP, DIP, IP-proto, L4SP, L4DP, DSCP, F3=packet-type)
- VLAN *QoS* - disabled by default - 1 slice, n rules (n VLANs)
  - Slice A or B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)
- port QoS - disabled by default - 1 slice, 1 rule
  - Slice D (F1=anything, F2=anything, F3=anything)
- *VRRP (Virtual Router Redundancy Protocol)* - 2 slices, 2 rules
  - Slice A (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=packet-type)
  - Slice A or B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)
- *EAPS (Extreme Automatic Protection Switching)* - 1 slice, 1 rule (master), n rules (transit - n domains)
  - Slice A or B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)
- *ESRP (Extreme Standby Router Protocol)* - 2 slices, 2 rules
  - Slice A (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=packet-type)
  - Slice A or B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)

- IPv6 - 2 slices, 3 rules
  - Slice A or B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)
  - Slice (F1=Port-list, F2=DIPv6, IPv6 Next Header Field, TC, F3=anything)
- Netlogin - 1 slice, 1 rule
  - Slice A or B (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)
- VLAN Mirroring - 1 slice, n rules (n VLANs)
  - Slice E (F1=Port-list, F2=MACDA, MACSA, Etype, VID, F3=anything)
- Unicast Multiport *FDB*
  - 1 slice, 1+n rules in 24 port ExtremeSwitching series switches
  - 1 slice, 2+ n rules in 48 port ExtremeSwitching series and G48Ta, G48Pe cards
- VLAN Aggregation
  - 1 slice, 4 rules for the first subvlan configured and 1 slice, 2 rules for subsequent subvlan configuration
- Private VLAN
  - 2 slices, 3 rules when adding an non-isolated VLAN with loop-back port a to private VLAN
  - 1 slice, 3 rules when adding an isolated subscriber VLAN (without loopback port) to a private VLAN. 3 additional rules when a loopback port is configured in the above isolated subscriber VLAN
- ESRP Aware - 1 slice, 1 rule
  - Field 1: {Drop, OuterVlan, EtherType, PacketFormat, HiGig, Stage, StageIngress, Ip4, Ip6}
  - Field 2: {SrcIp, DstIp, L4SrcPort, L4DstPort, IpProtocol, DSCP, Ttl, Ip6HopLimit, TcpControl, IpFlags}
  - Field 3: {RangeCheck}
- *ACL* rule with mirror action is installed in a separate slice, and this slice cannot be shared by other rules without a mirror action.

Note
The user ACLs may not be compatible with the slice used by this ESRP rule. This may result in the reduction the number of rules available to the user by 127.

To display the number of slices used by the ACLs on the slices that support a particular port, use the following command:

```
show access-list usage acl-slice port port
```

To display the number of rules used by the ACLs on the slices that support a particular port, use the following command:

```
show access-list usage acl-rule port port
```

To display the number of Layer 4 ranges used by the ACLs on the slices that support a particular port, use the following command:

```
show access-list usage acl-range port port
```

System Configuration Example

The following example shows incremental configurations and their corresponding *ACL* resource consumption taken on a switch.

- Default configuration including: dot1p examination and *IGMP* snooping:
  - 2 slices, 10 rules
- Add an IP interface to the configuration:
  - 2 slices, 13 rules
- Add port-based *QoS* to the configuration:
  - 2 slices, 14 rules
- Add *VLAN*-based QoS to the configuration:
  - 2 slices, 15 rules
- Add *VRRP* to the configuration:
  - 2 slices, 17 rules
- Add *EAPS* (Master mode) to the configuration:
  - 2 slices, 18 rules
- Add *ESRP* to the configuration:
  - 2 slices, 21 rules
- Add IPv6 routing (slowpath) to the configuration:
  - 4 slices, 24 rules
- Add Netlogin to the configuration:
  - 5 slices, 25 rules

ACL Error Messages

Errors may happen when installing an *ACL* policy on a port, *VLAN*, or all interfaces (wildcard). Following is a list of the most common error conditions and their resulting CLI error message:

- `Error: ACL install operation failed - slice hardware full for port 3:1`

  Slice resource exceeded: This happens when all slices are allocated for a given chip and an additional incompatible rule (see Egress ACLs on page 395) is installed which requires allocation of another slice.

- `Error: ACL install operation failed - rule hardware full for port 3:1`

  Rule resource exceeded: This happens when all slices are allocated for a given chip and there is an attempt to install a compatible rule to the lowest precedence slice which already has 128 rules. This condition can be triggered with less than the full capacity number of rules installed. For example, if 15 of the slices each have less than 128 rules and there is an attempt to install 129 compatible rules, this error message will be displayed.

- `Error: ACL install operation failed - layer-4 port range hardware full for port 3:1`

  Layer-4 port range exceeded: This happens when more than 32 Layer 4 port ranges are installed on a single chip.

- `Error: ACL install operation failed - conditions specified in rule "r1" cannot be satisfied by hardware on port 3:1`

  Incompatible fields selected: This happens when the selected conditions can not be satisfied by the available single-slice field selections described in Compatible and Conflicting Rules on page 392.

- `Error: ACL install operation failed - user-defined-field (UDF) hardware full for port 3:1`

  UDF exceeded: This happens in the rare case that the two available user-defined fields are exceeded on a given chip. UDF fields are used to qualify conditions that are not natively supported by the hardware. Some ACL rules that use UDF are: Source MAC address + Destination IP address combination, Destination MAC address + Source IP address combination, *ICMP* Type, and ICMP Code.

ACL Counters-Shared and Dedicated

You can configure rule compression in *ACL*s to be either shared or dedicated.

In the dedicated mode, ACL rules that have counters are assigned a separate rule space and the counter accurately shows the count of matching events. If the ACL with counter is applied to ports 1 and 2, and 10 packets ingress via port 1 and 20 packets ingress via port 2, the ACL counter value for ports 1 and 2 is 10 and 20 packets respectively. More space is used and the process is slower than shared. Dedicated is the default setting.

In the shared mode, ACL space is reused even with counters. ACL counters count packets ingressing via all ports in the whole unit. If the ACL with the counter is applied to ports 1 and 2, and 10 packets ingress via port 1, and 20 packets ingress via port 2, the ACL counter value is 30 each of ports 1 and 2 instead of 10 and 20. The process is faster—as fast as applying an ACL without the counters—and saves space.

Note
Port-Counter shared mode will not work when ports are on across slots and units.

The shared/dedicated setting is global to the switch; that is, the option does not support setting some ACL rules with shared counters and some with dedicated counters.

Use the following command to configure the shared or dedicated mode:

```
configure access-list rule-compression port-counters [shared | dedicated]
```

Use the following command to view the configuration:

```
show access-list configuration
```

The shared or dedicated mode does not affect any ACLs that have already been configured. Only ACLs entered after the command is entered are affected.

Note
To configure all ACLs in the shared mode, enter the command before any ACLs are configured or have been saved in the configuration when a switch is booted.

Policy-Based Routing

Note
Policy-Based Routing is available only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document. Refer to Load Sharing Rules and Restrictions for All Switches on page 280 for information on applying ACLs to *LAG (Link Aggregation Group)* ports.

Layer 3 Policy-Based Redirect

Policy-Based Routing allows you to bypass standard Layer 3 forwarding decisions for certain flows. Typically, in a Layer 3 environment, when an IP packet hits an Ethernet switch or router, the Layer 3 processing determines the next-hop and outgoing interface for the packet (based only on the packet's destination address). The Layer 3 processing does so by looking up the IP Forwarding Table; this forwarding table itself is populated either by static routes or by routes learned dynamically from routing protocols such as *OSPF* and *RIP*.

With Policy-Based Routing, you can configure policies to use a different next-hop than what the routing lookup would have chosen. The switch first compares packets to the *ACL* rule entries. If there is a match, the packet is forwarded to the destination identified by the redirect action modifier. If there is no match, the packet is forwarded based on normal routing, in other words, by looking up a route in the IP Forwarding Table.

When there is a match with a redirect ACL rule, the matched traffic is redirected to the next-hop specified in the action.

Note
The IP packet itself is not modified, but only redirected to the port where the next-hop entry resides. The original IP destination address and source address are retained in the packet. The TTL is decremented and the IP checksum is recalculated.

The applications for Policy-Based Routing are quite diverse, since the functionality can be used to set policies on how flows identified by any Layer 2 to Layer 7 field (bounded by the switch's ACL syntax) are forwarded.

Deployment scenarios include:

- Forwarding flows for certain applications, for example, all HTTP traffic to designated server(s).

- Redirecting flows from certain source IP addresses for security and other applications.

Policy-Based Routing is implemented using ACLs, so it inherits the capabilities and limitations of ACLs. All the matching conditions used for ACLs can be used for Policy-Based Routing. The destination IP address must be an IPv4 unicast address. For IPv6 scenarios refer the section on Policy-Based Redirection Redundancy on page 406.

When a switch finds a matching ACL rule, it forwards the packet to the redirect IP address as specified in the rule without modifying the packet (except as noted above).

The traffic flow is redirected only after applying the ACL to the port and only when the redirect IP address's adjacency is resolved. When the ARP or NDP table does not have the information to reach the redirect IP address, the packet is routed based on the Layer 3 routing table. When the switch does not know how to reach the redirect IP address in the rule, the rule is installed with a warning, and traffic is not redirected until the address is resolved in the ARP or NDP table. After the address is resolved, the traffic is redirected.

To configure Policy-Based Routing, you configure an ACL on your switch. You can apply an ACL policy file, or use a dynamic ACL.

The following is an example ACL rule entry that redirects any TCP traffic with a destination port of 81 to the device at IP address 3.3.3.2:

```
entry redirect_port_81 {
    if {
        protocol tcp;
        destination-port 81;
    } then {
        redirect 3.3.3.2;
    }
}
```

Use the following procedure:

1. Issue the following command to prevent the redirect IP address from clearing from the ARP or NDP table due to a timeout: `enable iparp refresh`
2. Configure the ACL, either by applying an ACL policy file similar to the example, or a dynamic ACL.
3. Ping or send traffic so that the redirect IP adjacency is resolved.

You may want to create a static ARP or NDP entry for the redirect IP address, so that there will always be a cache entry.

Note
An ACL can be rejected on switches that support Policy-Based Routing, because these have different amounts of hardware resources and one switch has exhausted its resources.

Layer 2 Policy-Based Redirect

This feature allows matching packets to override the normal forwarding decision and be Layer 2 switched to the specified physical port. This is accomplished using an additional packet *ACL* lookup. While similar to the Layer 3 Policy-Based Redirect feature, it differs in that the packet is not modified for Layer 3 routing based on a new IP redirect next-hop. Instead, the packet uses the packet format based on the forwarding decision. When the packet is Layer 2-switched, the packet egresses the redirect port unmodified. When the packet is Layer 3-switched, the packet egresses with the Layer 3 packet modifications of the next-hop found by the normal Layer 3 forwarding lookups.

The following ACL actions are added in support of this feature:

`redirect-port` *`port`*`; redirect-port-list` *`port-list`*

Note

The `redirect-port` or `redirect-port-list` commands will not work for L3 switched packets matching ACL, if distributed IP ARP feature is turned ON.

You must specify the *`port`* argument in the correct format for the switch platform. On supporting switches, this argument must be in the format *`slot`*:*`port`* and this argument must be in the format *`port`*.

The *`port-list`* argument is simply a comma-separated list of *`port`* arguments. White space between *`port`* arguments is not allowed.

Here is an example of valid *`port-list`* syntax:

```
redirect-port-list 2:1,2:5,5:3; and redirect-port-list 3,24,5;
```

Here is an example of **invalid** *`port-list`* syntax :

```
redirect-port-list 2:1 2:5 5:3;
redirect-port-list 2, 4, 5;
```

The policy shown below redirects any TCP traffic with source Layer 4 port 81 to physical port 3:2.

```
entry one {
    if {
        protocol tcp;
        source-port 81;
        destination-port 200 ;
    } then {
        count num_pkts_redirected;
        redirect-port 3:2;
    }
```

The policy shown below redirects any in-profile traffic as defined by the meter configuration to physical port 14. The out-of-profile traffic is subject to the action specified in the meter “out-action” configuration.

```
entry one {
    if {
```

```
    } then {
        meter redirected_traffic;
        count num_pkts_redirected;
        redirect-port 14;
    }
```

The policy shown below redirects all traffic with source IP matching 192.168.1.1/24; to physical ports 2:10 and 4:7.

```
entry one {
if {
source-address 192.168.1.1/24;
} then {
count num_pkts_redirected;
redirect-port-list 2:10,4:7;
}
```

If an incorrect port format is used or if the port number specified is out of range, the following error message is displayed:

```
switch.5 # check policy l2pbr
Error:  Policy l2pbr has syntax errors
Line 7 : 12:3 is not a valid port.
switch.6 # check policy l2pbr
Error:  Policy l2pbr has syntax errors
Line 7 : 77 is not a valid port.
```

When this feature is used on ExtremeSwitching switches, the traffic egressing the redirect-port can be either tagged or untagged depending on the redirect-port *VLAN* configuration. The following table provides the details.

Table 30: VLAN Format of Traffic Egressing Redirect-Port

ACL Hardware Type	Redirect-Port Not in Egress VLAN	Redirect-Port Tagged in Egress VLAN	Redirect-Port Untagged in Egress VLAN
ExtremeSwitching Switches	Dropped	VLAN Tagged	Untagged

Be aware of the following important implementation notes:

- Using the “redirect-port” action with a disabled port causes traffic to be dropped.
- For systems with a- and e- series hardware that has the larger table size, packets with IP options do not match ACLs using the “redirect-port” action. Systems with hardware that has the smaller table size do not have this capability. On these systems, packets with IP options will match ACLs that use the “redirect-port” action, and will be dropped.
- The redirect-port-list action modifier is targeted towards L2 scenarios. This action is not supported in slow path ACLs. The following list summarizes the behavior of the redirect-port-list action modifier under certain situations.

The following list summarizes the behavior of redirect-port-list action modifier under certain situations.

- When a Unicast packet matches the applied ACL, the packet is redirected to all ports specified in the redirect port-list as long as these ports are part of the true egress VLAN.
- When a Broadcast/Multicast packet matches the applied ACL, the packet is redirected only to ports specified in the redirect port-list that are part of the ingress VLAN. Matched multicast packets will get L2 switched.
- When a *LAG* port is part of redirect-port-list, then packets matching applied ACL will be load shared between LAG member ports based on Layer 2 source and destination MAC addresses.

LAG Port Selection

This feature allows you to apply an *ACL* that causes matching packets to egress a specific port in a link aggregation (or load-sharing) group.

The following ACL action is added in support of this feature:

```
redirect-port-no-sharing port
```

The ACL overrides any load-sharing algorithm hash that is generated based on the lookup results.

Limitations include the following:

- If the selected port in a load-sharing group is down, the packets will be dropped.
- Like the redirect-port action, the specified port must be a member of the egress *VLAN*.

Following is an example of a configuration and ACL policy that directs traffic matching 10.66.4.10 to LAG port 3:

```
enable sharing 2 group 2,3
radiomgmt.pol:
entry one {
if {
destination-address 10.66.4.10/32;
} then {
redirect-port-no-sharing 3;
}
}
config access-list radiomgmt any
```

This example would direct inband management traffic to specific radios connected to specific ports within a load-sharing group.

Policy-Based Redirection Redundancy

Multiple Next-hop Support

As discussed above, Layer 3 and Layer 2 policy-based redirect support only one next-hop for one policy-based entry. Multiple next-hops with different priorities can be

configured. A higher priority is denoted with a higher number; for example, "priority 5" has a higher precedence than "priority 1." When a high priority next-hop becomes unreachable, another preconfigured next-hop, based on priority, replaces the first. This is done by first creating a flow-redirect name that is used to hold next-hop information. User-created flow-redirect names are not case-sensitive.

Note
As of ExtremeXOS 16.1, there is no limitation in creating the flow-redirects. Number of Next hops has been increased to 4,096 next hops. If more than 4,096 next hops are attempted to be created, an error message appears.

Use the following command:

```
create flow-redirect flow_redirect_name
```

To delete the flow-redirect name, use:

```
delete flow-redirect flow_redirect_name
```

Then information for each next-hop, including a defined priority, is added one by one to the new flow-redirect name. Use the following command:

```
configure flow-redirect flow_redirect_name add nexthop ipaddress
priority number
```

Note
You can add IPv4 or IPv6 next-hops to a flow-redirect policy, but both types are not supported in the same policy.

To delete a next-hop, use the following command:

```
configure flow-redirect flow_redirect_name delete nexthop {ipaddress |
all }
```

Because an *ACL* does not recognize the virtual routing concept, one policy-based routing is used for multiple virtual routing entries when a *VLAN*-based is used for one port. Configuring a virtual router into a flow-redirect allows policy-based routing to work for only one specific virtual router. Use the following command:

```
configure flow-redirect flow_redirect_name vr vr_name
```

Note
Flow-redirect does not work on user-created virtual routers.

Finally, a new action modifier, redirect-name, is used to specify the flow-redirect name in an ACL rule entry.

```
entry redirect_redundancy {
    if match all {
        source-address 1.1.1.100/24 ;
    } then          {
        permit ;
        redirect-name <name>
    }
}
```

Health Checking for ARP, NDP, and Ping

Policy-based redirection redundancy requires the determination of the reachability or unreachability of the active next hop and the other configured next hops. This feature can use ARP, NDP or Ping checking to make the determination.

Note
IPv6 Policy-Based Routing is not supported for traffic with Hop-by-Hop extension headers. Traffic will continue to be hardware forwarded, and will not be processed in slow path.

To configure health checking for a specific flow-redirect-name, use the following command:

```
configure flow-redirect flow_redirect_name health-check [ping | arp |
neighbor-discovery]
```

To configure the ping interval, miss count, and success count for a next-hop, use the following command:

```
configure flow-redirect flow_redirect_name nexthop ip_address ping
health-check interval seconds miss number {success successes}
```

Packet Forward/Drop

The default behavior for policy-based routing when all next-hops are unreachable is to route packets based on the routing table. Policy-based routing redundancy adds an option to drop the packets when all next-hops for the policy-based routing become unreachable.

To configure this option, use the following command:

```
configure flow-redirect flow_redirect_name no-active [drop|forward]
```

Configuring Packet Forward Drop

Traffic from the Source IP = 211.10.15.0/24, 211.10.16.0/24 network blocks should be redirected into two routers: 192.168.2.2 and 192.168.2.3. The 192.168.2.2 router is preferred to 192.168.2.3. If router 192.168.2.2 is not reachable, 192.168.2.3 should be used. If both routers are not reachable, the default route is used.

1. Create a flow-redirect to keep next-hop IP address and health check information.

   ```
   create flow-redirect premium_subscriber
   config flow-redirect premium_subscriber add next-hop 192.168.2.2
   priority 200
   config flow-redirect premium_subscriber add next-hop 192.168.2.3
   priority 100
   ```

2. Add an ACL entry with a flow-redirect name action to the existing ACL policy

   For example: premium_user.pol:

   ```
   entry premium_15 {
   if match {
   source-address 211.10.15.0/24;
   } then {
   ```

```
    permit;
    redirect-name premium_subscriber;
    }
    }
    entry premium_16 {
    if match {
    source-address 211.10.16.0/24;
    } then {
    permit;
    redirect-name premium_subscriber;
    }
    }
```

3. Apply the modified ACL policy file or dynamic ACL into a port, VLAN, or VLAN and Port.

   For example: user1 VLAN: 192.168.1.0/30, user2 VLAN: 192.168.1.4/30.

```
    config access-list premium_user vlan user1 ingress
    config access-list premium_user vlan user2 ingress
```

4. Finally, check the current flow-redirect status.

```
BD-8810.47 # show flow-redirect "premium_subscriber"
Name         : premium_subscriber           VR Name         : VR-Default
NO-ACTIVE NH : FORWARD                      HC TYPE         : PING
NH COUNT     : 2                            ACTIVE IP       : 192.168.2.3
Index     STATE     Pri          IP ADDRESS              STATUS INTERVAL MISS
==========================================================================
0         ENABLED   200          192.168.2.2              DOWN
2
           2
1         ENABLED   100          192.168.2.3              UP       2      2

BD-8810.48 # show flow-redirect
Flow-Redirect Name     NH_CNT     ACTIVE IP    VR Name         D/F  HC
=========================================================================
premium_subscriber     2          192.168.2.3  VR-Default       F   PING
```

ACL Troubleshooting

The following commands are designed to help troubleshoot and resolve *ACL* configuration issues:

- `show access-list usage acl-mask port` *`port`*
- `show access-list usage acl-range port` *`port`*
- `show access-list usage acl-rule port` *`port`*
- `show access-list usage acl-slice port` *`port`*

The acl-mask keyword is not relevant for the a-series or e-series models.

If you enter this command and specify an a-series or e-series port, the following error message appears:

```
This command is not applicable to the specified port.
```

Use the acl-rule keyword to display the total number of ACL rules that are available and consumed for the specified port.

If this keyword is specified on an a-series or e-series port, the first part of the command output details the port list using this resource because the ACL hardware rules are shared by all ports on a given ASIC (24x1G ports). If you enter the same command and specify any of the listed ports, the command output is identical.

```
*switch# show access-list usage acl-rule port 4:1 Ports 4:1-4:12, 4:25-4:36
Total Rules:     Used: 46  Available: 2002
```

The **acl-slice** keyword is used to display ACL resource consumption for each of the independent TCAMs, or slices, that make up the hardware ACLs.

Each slice is a 128-entry TCAM. The command output displays the number of consumed and available TCAM rules for each slice as follows.

```
*switch# show access-list usage acl-slice port 4:1
Ports 4:1-4:12, 4:25-4:36
Slices:          Used: 8  Available: 8
Slice 0 Rules:   Used: 1  Available: 127
Slice 1 Rules:   Used: 1  Available: 127
Slice 2 Rules:   Used: 1  Available: 127
Slice 3 Rules:   Used: 8  Available: 120
Slice 4 Rules:   Used: 8  Available: 120
Slice 5 Rules:   Used: 2  Available: 126
Slice 6 Rules:   Used: 1  Available: 127
Slice 7 Rules:   Used: 24 Available: 104
```

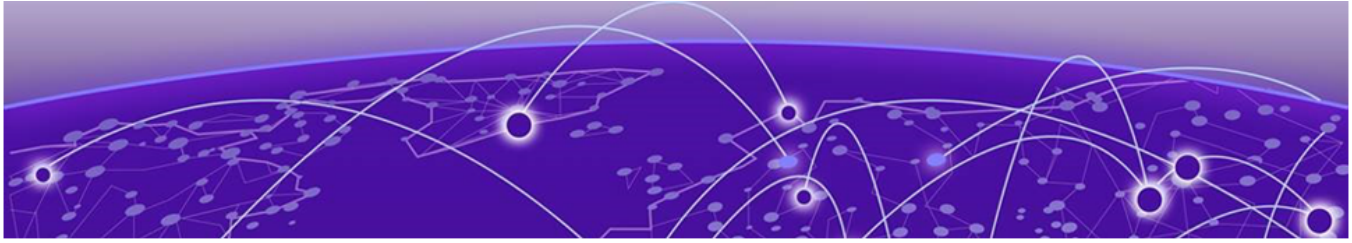
Use the **acl-range** keyword to view the Layer-4 port range hardware resource on an a-series or e-series model switch.

Each a-series and e-series ASIC has 16 Layer-4 port range checkers that are shared among the 24 1G ports. The first part of the command output lists the ports that utilizes this resource. The second part of the command output lists the number of range checkers that are consumed and the number available for use.

```
switch # show access-list usage acl-range port 4:1
Ports 4:1-4:12, 4:25-4:36
L4 Port Ranges:  Used: 0  Available: 16
```

If the **acl-slice** or **acl-range** keyword is specified with an e-series port, the following error message will appear:

```
This command is not applicable to the specified port.
```



AVB

Overview on page 411
AVB Feature Pack License on page 412
Configuring and Managing AVB on page 412
Displaying AVB Information on page 417

This chapter provides information about Audio Video Bridging support. It specifically discusses the AVB Feature Pack License, as well as how to configure and manage the AVB feature.

Overview

Audio Video Bridging (AVB) supports the deployment of professional quality audio and/or video (AV) over standard Ethernet while coexisting with other "legacy" (or non-AV) Ethernet traffic. This supports "Network Convergence," or using one simple standard Ethernet network for all communication needs.

To support AV applications, it is necessary for AVB systems to provide time synchronization and *QoS (Quality of Service)*.

Time Synchronization is needed so that multiple streams may be synchronized with respect to each other. For example:

- Voice and video
- Multiple audio streams for a multi-digital speaker deployment in a large venue
- Multiple Video streams in a security surveillance application

QoS is needed to ensure:

- Bandwidth guarantees sufficient for each application
- Worst Case Delay Bounds, particularly for interactive applications
- Traffic shaping to limit traffic burstiness and reduce buffering requirements

The time synchronization and QoS requirements for AVB systems are defined in the following set of IEEE Standards:

- IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks (gPTP)
- IEEE 802.1Q
  - Clause 10: Multiple Registration Protocol (MRP) and Multiple MAC Registration Protocol (MMRP)
  - Clause 11: *VLAN (Virtual LAN)* Topology Management (MVRP)

  - Clause 34: Forwarding and Queuing for Time-Sensitive Streams (FQTSS)
  - Clause 35: Stream Reservation Protocol (SRP)
- IEEE 802.1BA: Audio Video Bridging (AVB) Systems

AVB Feature Pack License

All ExtremeSwitching Universal switches have the AVB Feature License included in the Base License. You do not need to separately purchase and install the AVB Feature License for these switches.

For more information about the AVB Feature Pack, supported platforms, and licensing, see Switch Engine v33.6.1 Licensing Guide .

Note
AVB is not supported on SummitStacks.

Note
AVB is not supported when MACsec is enabled.

Configuring and Managing AVB

AVB is not enabled in the default configuration, and must be enabled both globally on the switch and on the ports where you want to use it. In the simplest case, when starting with a blank configuration, AVB may be enabled by executing the following two commands:

```
# enable avb

enable avb ports all
```

The status of AVB can be seen by using the following command:

```
# show avb
gPTP status          : Enabled
gPTP enabled ports   : *1s     *2m    *10m    *11m    *12m
                       *13m    *14m   *15m    *16m    *17m
                       *18m    *19m   *20m    *21m

MSRP status          : Enabled
MSRP enabled ports   : *1ab    *2ab   *10ab   *11ab   *12ab
                       *13ab   *14ab  *15ab   *16ab   *17ab
                       *18ab   *19ab  *20ab   *21ab

MVRP status          : Enabled
MVRP enabled ports   : *1      *2     *10     *11     *12
                       *13     *14    *15     *16     *17
                       *18     *19    *20     *21

Flags:         (*) Active,                   (!) Administratively disabled,
               (a) SR Class A allowed,       (b) SR Class B allowed,
               (d) Disabled gPTP port role,  (m) Master gPTP port role,
               (p) Passive gPTP port role,   (s) Slave gPTP port role
```

The `show avb` command displays high level information about each of the three main protocols (gPTP, MSRP, and MVRP). Each protocol section indicates that all three protocols are enabled both globally, and on ports 1,2 and 11-21. The “*” indicates that we have link on each of the ports.

The gPTP status indicates that port 1 is a slave port, which means that the Grand Master Clock (GMC) is reachable through port 1. The gPTP status also indicates that the rest of the ports are master ports. Furthermore, the fact that no ports are shown to be in the Disabled role means that gPTP is operational on all the ports.

ExtremeXOS provides static AVB configuration with the exception of of Best Master Clock Algorithm (BMCA), which runs as part of gPTP. The BMCA feature adds the ability to disable BMCA and to specify a slave port if desired. When BMCA functionality is on, BMCA is executed normally as described in IEEE 802.1AS. When BMCA is off, BMCA is not executed. In disabled mode, a port can be configured to be the slave-port. If no ports are configured to be slave-port, then the switch will become the Grandmaster Clock and all network-gptp enabled ports will be master ports. The `show network-gptp` command displays whether BMCA is on or off.

The "ab" on the MSRP status indicates that all ports are members of both the class A and class B domain domains. The MVRP status simply shows which ports are enabled and active.

The user interface for AVB includes the following five protocols:

- gPTP
- MRP
- MVRP
- MSRP
- FQTSS

The "avb" commands shown above are part of a set of AVB macro commands provided to simplify the process of enabling and disabling AVB. The AVB macro commands have the form:

```
[ enable | disable | unconfigure ] avb { ports [ all | port_list ] }
```

Using one of the macro commands is the same as executing the following three commands:

```
[ enable | disable | unconfigure ] network-clock gptp { ports [ all |
port_list ]}

[ enable | disable | unconfigure ] mvrp { ports [ all | port_list ]}

[ enable | disable | unconfigure ] msrp { ports [ all | port_list ]}
```

MRP does not need to be enabled or disabled, and the only MRP properties that may be configured are timer values. The defaults should be sufficient for most deployments, though it may be necessary to increase the leave-all and leave time values when supporting a large number of streams.

Link aggregation allows an increased bandwidth and resilience by using a group of ports to carry traffic in parallel between switches. Multiple ports can be aggregated into one logical port. MVRP can be enabled on the logical port. The MVRP control packets will be transmitted on any available physical port of the *LAG (Link Aggregation Group)*. The peer on the other side will receive the packet and process as if being received on the logical port. MVRP supports both dynamic (LACP) as well as static load sharing. Some restrictions that apply are:

- The individual ports of the LAG, including the master port, should not have MVRP configuration prior to grouping.
- MVRP can be enabled / disabled only on master port. The individual links cannot be configured.
- Once sharing is disabled, MVRP configuration of the master port will be lost (default will be disabled).
- The statistics and counters shown on the MVRP show commands will be a cumulative counter for all links added together. We do not maintain per link counters.
- The actual load sharing of the traffic is beyond MVRP's domain and should take place as per the configured LAG setting. MVRP just adds the LAG port to the VLAN(s).
- MVRP only works with Rapid Spanning Tree Protocol [RSTP} and the Default VLAN.User-created STPD instances cannot be associated with the Default VLAN. The Default VLAN is always associated to STPD s0.

MVRP data structure is based on port Instance. All dynamic VLANs created or propagated for a given port will be stored for each port Instance. For normal ports, the port Instance will correspond to the PIF port instance, and for LAG ports, the port Instance will correspond to the LIF port Instance. The port instance is not shown in any of the standard CLI show commands, though it is available as a part of the debug commands. Once MVRP is enabled on the master port, addition / deletion of individual links is supported. MVRP packets received on the newly added link will be accounted instantaneously.

MVRP LAG configuration examples:

- `enable sharing 13 grouping 13,14,15`
- `enable mvrp port 13`
- `enable sharing 13 grouping 13,14`
- `enable mvrp port 13`
- `configure sharing 13 add ports 15`

The FQTSS settings are managed by MSRP, and may not be configured directly.

The disable commands disable the AVB protocols globally or per port without changing any other configured settings, while the unconfigure commands reset all AVB settings to the initial states, and release all switch resources that were allocated when the protocols were enabled.

More detailed configuration options are provided on a per-protocol basis using the corresponding configure commands:

```
configure network-clock gptp
configure mvrp
configure msrp
configure mrp
```

If a topology contains redundant connections, *ELRP (Extreme Loop Recovery Protocol)* with dynamic VLAN feature should be enabled to prevent data loops in VLANs created by MVRP (see ELRP with Dynamic VLANs on page 2121). The ELRP with dynamic VLANs feature is supported starting with ExtremeXOS 22.2. In a MVRP-only environment, STP-based loop prevention method is not supported.

MRP/MSRP/MVRP LAG Support

As of ExtremeXOS 21.1, *LAG* is supported for MRP/MSRP. A LAG may have one or multiple member ports, thus providing redundancy as well as additional bandwidth.

There are two modes of how a LAG runs with MRP/MSRP. The two MRP/MSRP LAG modes are single-port and cumulative. The concept of effective bandwidth is used below to simulate the available bandwidth of a physical port (port speed). As in the case of the physical port, a configurable percentage, deltaBandwidth, which by default is 75%, of the effective bandwidth is the final bandwidth available for MSRP streams.

In single-port mode, the LAG is simply a way to provide redundancy. Therefore the effective bandwidth is set to the minimum bandwidth of all member ports.

```
effective bw = min (bw of all active member ports)

MSRP reservable bw = deltaBandwidth * effective bw
```

All LAG member ports on ExtremeXOS must have the same speed to aggregate. Therefore, the minimum bandwidth is equivalent to the bandwidth of any member port.

In cumulative mode, the LAG trades redundancy for extra bandwidth. To calculate the available bandwidth, all bandwidth of one active member and partial bandwidth of other member ports are used to calculate the effective bandwidth.

```
effective bw = min (bw of all active member ports) + beta * sum (bw of
all other active member ports)

MSRP reservable bw = deltalBandwidth * effective bw
```

For example, if the LAG has three 1GB member ports, and deltaBandwidth = 75%, then in the single-port mode, the effective bandwidth is 1GB, of which 75%, that is 0.75GB is reservable for MSRP, exactly the same as in the case of a physical port.

For the same LAG in the cumulative mode, if beta = 50%, then 100% of one member port and 50% bandwidth of other two member ports contributes to the effective bandwidth, therefore:

```
effective bw = 1GB (one active member port) + 50% * 1GB (the next active
member port) + 50% * 1GB (the last active member port) = 2GB

MSRP reservable bw = 75% x effective bw = 1.5GB
```

When a LAG runs in the cumulative mode, the streams are roughly evenly distributed on all member links. Even though beta percent bandwidth of the other member ports contributes to the effective bandwidth, it does not mean that reservation on that member port is limited to beta percent of its bandwidth. That is just the way to calculate an estimate of how much bandwidth can be provided with a LAG, with a balanced redundancy requirement. All member ports are treated equally and programmed in the same way.

As in the case of physical ports, if the `(total requested BW) <= deltaBandwidth * (effective BW)`, then assume the LAG is able to handle all streams safely and provide a comfortable level of redundancy if the streams are reasonably evenly distributed. In extremely polarized cases, where many streams are hashed to a certain member link, packet drop is inevitable. When `(total requested BW) > deltaBandwidth * (effective BW)`, then the LAG is not able to safely handle all streams, even though it may still be able to. At this time new reservation requests are declined.

The effective bandwidth is an aggregate of multiple physical ports and can exceed the bandwidth of any single physical port. The *QoS* profile is configured on a physical port, so if the reservation bandwidth request is less than the delta bandwidth of the physical port, the QoS profile will be configured to the requested bandwidth. To prevent the QoS profile configuration from exceeding the physical port bandwidth, the QoS profile configuration is limited to delta bandwidth of the physical port (which is typically 75% of the port bandwidth).

Multiple *VLAN* Registration Protocol (MVRP) is supported over (for more information, see Multiple VLAN Registration Protocol (MVRP) over Multi-switch Link Aggregation (MLAG) on page 290).

gPTP LAG Support

Starting in ExtremeXOS 21.1, configuration of gPTP on a *LAG* port is supported. However, the protocol is still running on the member port, aka, physical port level. When gPTP is enabled on a LAG port, all member ports are enabled with the protocol, vice versa when it is disabled on the LAG port. All member ports share the same gPTP configurations. When adding or deleting LAG member port, gPTP is automatically enabled or disabled. Member ports go operational up/down when the physical port is administratively enabled/disabled.

It is recommended to perform configuration on the LAG level; however, the only keyword in the `enable/disable/configure network-gptp` CLIs allows per member port fine tuning. This per member port change is overwritten with the next LAG level configuration. Unconfigure gPTP on a member port is not allowed though; otherwise, the newly added member port may not have the reference of the master port for gptp configuration.

In the CLI display, all member ports with gPTP enabled, regardless port operational status, are listed, but noted as a LAG member port. The LAG itself is not displayed because it does not participate the protocol.

Displaying AVB Information

The complete set of "show" commands are detailed in the Switch Engine v33.6.1 Command References . Some of the more commonly used commands are outlined here.

gPTP

Detailed information about gPTP can be displayed using the following set of commands:

```
show network-clock gptp ...
```

For example, the `show network-clock gptp ports` command can be used to view the gPTP properties of a given port, and is useful for debugging when the `show avb` command shows that the port is not operational for gPTP.

```
# show network-clock gptp ports 1
Physical port number  : 1
gPTP port status                    : Enabled
Clock Identity                      : 00:04:96:ff:fe:51:ba:ea
gPTP Port Number                    : 1
IEEE 802.1AS Capable                : Yes
Port Role                           : 9 (Slave)
Announce Initial Interval           : 0 (1.000 secs)
Announce Current Interval           : 0 (1.000 secs)
Announce Receipt Timeout            : 3
Sync Initial Interval               : -3 (0.125 secs)
Sync Current Interval               : -3 (0.125 secs)
Sync Receipt Timeout                : 3
Sync Receipt Timeout Interval       : 375000000 ns
Measuring Propagation Delay         : Yes
Propagation Delay                   : 623 ns
Propagation Delay Threshold         : 3800 ns (auto)
Propagation Delay Asymmetry         : 0
Peer Delay Initial Interval         : 0 (1.000 secs)
Peer Delay Current Interval         : 0 (1.000 secs)
Peer Delay Allowed Lost Responses   : 3
Neighbor Rate Ratio                 : 1.000020
PTP Version                         : 2
```

MSRP

Detailed information about MSRP can be displayed using the following set of commands:

```
show msrp ...
```

Several that are commonly used are:

```
show msrp
show msrp streams
```

```
show msrp listeners
show msrp streams propagation
```

Examples of these commands are shown below.

The `show msrp` command displays the summary information included in the `show avb` command, but also displays the total number of streams and reservations on the switch.

```
# show msrp
MSRP Status  : Enabled
MSRP Max Latency Frame Size   : 1522
MSRP Max Fan-in Ports         : No limit
MSRP Enabled Ports            : *1ab    *2ab    *10ab   *11ab    12
                                  13      14      15      16      17
                                  18      19      20      21
Total MSRP streams            : 2
Total MSRP reservations       : 6
Flags:  (*) Active,              (!) Administratively disabled,
        (a) SR Class A allowed,  (b) SR Class B allowed
```

The `show msrp` streams command displays all of the streams that the switch is aware of.

```
# show msrp streams
Stream Id  Destination     Port  Dec    Vid  Cls/Rn     BW
---------------------  -----------------  ----  ----  ----  ------  ---------
00:50:c2:4e:db:02:00:00  91:e0:f0:00:ce:00     1  Adv      2  A/1      6.336 Mb
00:50:c2:4e:db:06:00:00  91:e0:f0:00:0e:82     2  Adv      2  A/1      6.336 Mb

Total Streams: 2
--------------------------------------------------------------------------------
BW       : Bandwidth,                Cls      : Traffic Class,
Dec      : Prop. Declaration Types,  Rn       : Rank

MSRP Declaration Types:
   Adv   : Talker Advertise,         AskFail  : Listener Asking Failed,
   Fail  : Talker Fail,              RdyFail  : Listener Ready Failed,
   Ready : Listener Ready
```

The `show msrp listeners` command displays all of the listeners the switch is aware of. If the declaration type is either Ready or RdyFail, a reservation has been made, and the Stream Age will show the length of time this reservation has been active.

```
# show msrp listeners
     Stream Id            Port  Dec       Dir      State      Stream Age

                                                   App  Reg  (days,hr:mm:ss)
----------------------  -----  ------- -------  ---  ---  ----------------
00:50:c2:4e:db:02:00:00      2  Ready   Ingress  VO   IN       0, 01:40:23

                            10  Ready   Ingress  VO   IN       0, 01:27:05
                            11  Ready   Ingress  VO   IN       0, 01:27:05
00:50:c2:4e:db:06:00:00      1  Ready   Ingress  VO   IN       0, 01:40:15
                            10  Ready   Ingress  VO   IN       0, 01:27:05
                            11  Ready   Ingress  VO   IN       0, 01:27:05
--------------------------------------------------------------------------------
App      : Applicant State,               Dec     : MSRP Declaration Types
Dir      : Direction of MSRP attributes,  Reg     : Registrar State

MSRP Declaration Types:
AskFail  : Listener Asking Failed,        RdyFail : Listener Ready Failed,
```

```
Ready    : Listener Ready

Applicant States:
   AA    : Anxious active,                AN      : Anxious new,
   AO    : Anxious observer,              AP      : Anxious passive,
   LA    : Leaving active,                LO      : Leaving observer,
   QA    : Quiet active,                  QO      : Quiet observer,
   QP    : Quiet passive,                 VN      : Very anxious new,
   VO    : Very anxious observer,         VP      : Very anxious passive
Registrar States:
   IN    : In - Registered,               LV      : Leaving - Timing out
   MT    : Empty - Not Registered
```

The `show msrp streams propagation` command is useful for debugging the propagation of Talkers and Listeners for each stream.

```
# show msrp streams propagation stream-id 00:50:c2:4e:db:02:00:00
     Stream Id              Destination      Port  Dec   Vid   Cls/Rn     BW
----------------------   -----------------  ----  ----  ----  ------  ---------
00:50:c2:4e:db:02:00:00   91:e0:f0:00:ce:00     1  Adv      2    A/1    6.336 Mb


    Talker Propagation:
         Ingress     Ingress     Propagated        Propagated     Egress
         DecType     Port        DecType             Ports        DecType
         -------     -------     ----------        ----------     -------
        Adv     -->  1  -->      Adv        -->          2   -->
Adv
                                                      10   -->  Adv

                                                      11   -->  Adv
    Listener Propagation:
        Egress      Egress        Propagated        Listener       Ingress
        DecType     Port          DecType             Ports        DecType
        -------     -------      ----------        ----------      -------
        Ready   <--       1   <--  Ready      <--          2  <--  Ready

                              <--  Ready      <--         10  <--  Ready
                              <--  Ready      <--         11  <--  Ready
Total Streams: 1
------------------------------------------------------------------------------
BW       : Bandwidth,               Cls      : Traffic Class,
Dec      : Prop. Declaration Types, Rn       : Rank

MSRP Declaration Types:
   Adv   : Talker Advertise,         AskFail  : Listener Asking Failed,
   Fail  : Talker Fail,              RdyFail  : Listener Ready Failed,
  Ready  : Listener Ready
```

MVRP

Other than the MVRP summary information displayed in the show avb command, information about dynamically created VLANs is shown using the "vlan" commands as follows.

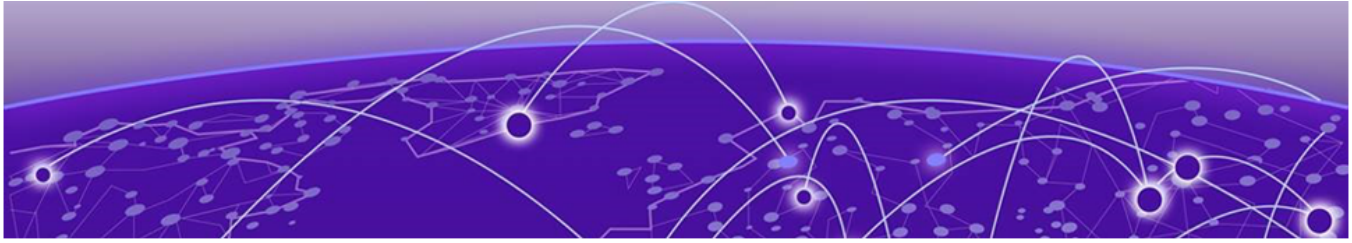
In the `show vlan` command, it can be seen that SYS_VLAN_0002 is a dynamically created *VLAN* due to the “d” flag.

```
# show vlan
------------------------------------------------------------------------------------------
---
```

```
Name  VID  Protocol  Addr       Flags                        Proto  Ports  Virtual

                                                                                Active
router
                                                                                /Total
-------------------------------------------------------------------------------------------
---
Default         1    ---------------------------------T--------------- ANY    4 /33  VR-
Default
Mgmt            4095 ---------------------------------------------- ANY    1 /1   VR-
Mgmt
SYS_VLAN_0002   2    ---------------------------------T------d-------- ANY    4 /4   VR-
Default
-------------------------------------------------------------------------------------------
---
Flags : (B) BFD Enabled, (c) 802.1ad customer VLAN, (C) EAPS Control VLAN,
        (d) Dynamically created VLAN, (D) VLAN Admin Disabled,
        (e) CES Configured, (E) ESRP Enabled, (f) IP Forwarding Enabled,
        (F) Learning Disabled, (i) ISIS Enabled,
        (I) Inter-Switch Connection VLAN for MLAG, (k) PTP Configured,
        (l) MPLS Enabled, (L) Loopback Enabled, (m) IPmc Forwarding Enabled,
        (M) Translation Member VLAN or Subscriber VLAN, (n) IP Multinetting Enabled,
        (N) Network Login VLAN, (o) OSPF Enabled, (O) Virtual Network Overlay,
        (p) PIM Enabled, (P) EAPS protected VLAN, (r) RIP Enabled,
        (R) Sub-VLAN IP Range Configured, (s) Sub-VLAN, (S) Super-VLAN,
        (t) Translation VLAN or Network VLAN, (T) Member of STP Domain,
        (v) VRRP Enabled, (V) VPLS Enabled, (W) VPWS Enabled,
        (Y) Policy Enabled, (Z) OpenFlow Enabled
       Total number of VLAN(s) : 3
```

You can display details about SYS_VLAN_0002 using the following command:

```
# show SYS_VLAN_0002
VLAN Interface with name SYS_VLAN_0002 created dynamically
    Admin State:      Enabled      Tagging:      802.1Q Tag 2
    Description:      None       Virtual router:   VR-Default
    IPv4 Forwarding:  Disabled
    IPv4 MC Forwarding:  Disabled
    IPv6 Forwarding:  Disabled
    IPv6 MC Forwarding:  Disabled
    IPv6:                None
    STPD:                   s0(Enabled)
    Protocol:            Match all unfiltered protocols
    Loopback:            Disabled
    NetLogin:            Disabled
    OpenFlow:            Disabled
    QosProfile:       None configured
    Flood Rate Limit QosProfile:       None configured
    Ports:   4.        (Number of active ports=4)
       Tag:       *1H,  *2H,   *10H,   *11H
         Flags:    (*) Active, (!) Disabled, (g) Load Sharing port
                   (b) Port blocked on the vlan, (m) Mac-Based port
                   (a) Egress traffic allowed for NetLogin
                   (u) Egress traffic unallowed for NetLogin
                   (t) Translate VLAN tag for Private-VLAN
                   (s) Private-VLAN System Port, (L) Loopback port
                   (e) Private-VLAN End Point Port
                   (x) VMAN Tag Translated port
                   (G) Multi-switch LAG Group port
                   (H) Dynamically added by MVRP
                   (U) Dynamically added uplink port
                   (V) Dynamically added by VM Tracking
```



BGP

BGP Overview on page 421
Configuring BGP on page 440
Managing BGP on page 452
Displaying BGP Information on page 454
Configuration Examples on page 455

The chapter is intended to provide information on the border gateway protocol. In this chapter you will find content on configuring and managing BGP, displaying BGP information, in addition to comprehensive BGP configuration examples.

BGP Overview

BGP (Border Gateway Protocol) is an exterior routing protocol that was developed for use in TCP/IP networks. The primary function of BGP is to allow different ASs to exchange network reachability information.

An AS is a set of routers that are under a single technical administration. This set of routers uses a different routing protocol, for example *OSPF (Open Shortest Path First)*, for intra-AS routing. One or more routers in the AS are configured to be border routers, exchanging information with other border routers (in different ASs) on behalf of all of the intra-routers.

BGP can be used as an exterior border gateway protocol (referred to as EBGP), or it can be used within an AS as an interior border gateway protocol (referred to as IBGP).

For more information on BGP, refer to the following documents:

- RFC 1745—BGP/IDRP for IP—OSPF Interaction
- RFC 1771—Border Gateway Protocol version 4 (BGP-4)
- RFC 1965—Autonomous System Confederations for BGP
- RFC 1966—BGP Route Reflection
- RFC 1997—BGP Communities Attribute
- RFC 2385—Protection of BGP Sessions via the TCP RSA Data Security, Inc. MD5 Message-Digest Algorithm Signature Option
- RFC 2439—BGP Route Flap Damping
- RFC 2545—Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
- RFC 2796—BGP Route Reflection - An Alternative to Full Mesh IBGP
- RFC 2918—Route Refresh Capability for BGP-4

- RFC 3392—Capabilities Advertisement with BGP-4
- RFC 4271—Border Gateway Protocol 4 (BGP-4)
- RFC 4360—BGP Extended Communities Attribute
- RFC 4456—BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)
- RFC 4486—Subcodes for BGP Cease Notification Message
- RFC 4724—Graceful Restart Mechanism for BGP
- RFC 4760—Multiprotocol Extensions for BGP-4
- RFC 4893—BGP Support for Four-octet AS Number Space
- RFC 5396—Textual Representation of Autonomous System (AS) Numbers
- draft_ieft_idr_restart_10.txt—Graceful Restart Mechanism for BGP

Note

ExtremeXOS supports BGP version 4 only, and does not support connections to peers running older versions of BGP.

For complete information about software licensing, including how to obtain and upgrade your license and what licenses are appropriate for these features, see the Switch Engine v33.6.1 Licensing Guide document..

BGP Four-Byte AS Numbers

The ExtremeXOS software supports 4-byte AS numbers, which can be entered and displayed in the ASPLAIN and ASDOT formats, which are described in RFC 5396, Textual Representation of Autonomous System (AS) Numbers.

Note

When entering an AS number in a policy file, you must enter a unique 2-byte or 4-byte AS number. The transition AS number, AS 23456, is not supported in policy files.

BGP Attributes

The following *BGP* attributes are supported by the switch:

- Origin—Defines the origin of the route. Possible values are Interior Gateway Protocol (IGP), Exterior Gateway Protocol (EGP), and incomplete.
- AS_Path—The list of ASs that are traversed for this route. The local AS-path is added to the BGP update packet after the policy is applied.
- AS4_Path—This attribute is used by 4-byte peers when sending updates to 2-byte peers. This attribute carries AS-number information that can be represented only in 4-bytes.
- Next_hop—The IP address of the next hop BGP router to reach the destination listed in the NLRI field.
- Multi_Exit_Discriminator—Used to select a particular border router in another AS when multiple border routers exist.
- Local_Preference—Used to advertise this router's degree of preference to other routers within the AS.

- Atomic_aggregate—Indicates that the sending border router has used a route aggregate prefix in the route update.
- Aggregator—Identifies the BGP router AS number and router ID for the router that performed route aggregation.
- AS4_Aggregator: This attribute is used by 4-byte peers when sending updates to 2-byte peers. This attribute carries AS-number information that can be represented only in 4-bytes.
- Community—Identifies a group of destinations that share one or more common attributes.
- Cluster_ID—Specifies a 4-byte field used by a route reflector to recognize updates from other route reflectors in the same cluster.
- Originator_ID—Specifies the router ID of the originator of the route in the local AS.
- Extended Community—Provides a mechanism for labeling BGP-4 update messages that carry information
- Multiprotocol reachable NLRI—Used to advertise a feasible BGP route for the non IPv4-unicast address family
- Multiprotocol unreachable NLRI—This attribute is used to withdraw multiple unfeasible routes from service

BGP Community Attributes

A *BGP* community is a group of BGP destinations that require common handling. ExtremeXOS supports the following well-known BGP community attributes:

- no-export
- no-advertise
- no-export-subconfed

Extended Community Attributes

The extended community attribute provides a mechanism to label *BGP* routes. It provides two important enhancements over the standard community attribute:

- An expanded range. Extended communities are 8-bytes wide, whereas regular communities were only 4-bytes wide. So, this ensures that extended communities can be assigned for a plethora of uses, without the fear of overlap.
- The addition of a 'Type' field provides structure for the extended community

The following two types of extended communities are available:

- Route Target (RT)
- Site Of Origin (SOO)

Although these two community types are generally used in L3 VPN network setup, you can also use them in a non-L3 VPN network to control the distribution of BGP routes.

BGP does not send either the extended or standard community attributes to their neighbors by default; you must use the configuration command `configure bgp neighbor send-community`.

Extended Community Processing

When *BGP* receives the extended community attribute in a route from its neighbor, it validates the community syntax. If the community is syntactically valid, the inbound neighbor route-policy is applied to the route. The inbound route-policy may contain extended-community statements in match block (in other words, an or/and set) of the policy. If the route is not rejected by the inbound route-policy, it is added to the LocRIB of the BGP along with the extended community. The **`detail`** option of the show bgp routes command displays the routes with the extended community attribute if they are present in that route's path attribute.

Associating the Extended Community Attribute to the BGP Route

The extended community attribute can be added to or removed from a *BGP* route using an ExtremeXOS policy in the same way this action is performed for a regular community attribute.

The extended-community keyword has been added in the Policy Manager, and can be used in the match as well as in the set block of a policy file.

Syntax in Match block

```
extended-community "<extended-community-1> <extended-community-2> …"
```

Where, the syntax of <extended-community-N> is

```
[rt|soo]:[<2-byte AS num>:<4-byte num> | <4-byte IP Address>:<2-byte num> | <4-byte AS
num>L:<2-byte num> | <first two bytes of AS num>.<last two bytes of AS num>:<2-byte num>]
```

The attributes are defined as follows:

- **`rt`**: route target extended community type
- **`soo`**: site of origin extended community type
- **`<2-byte AS number>`**: This is a 2-byte AS number; the use of private AS-number is not recommended
- **`<4-byte num>`**: a 4-byte unsigned number
- **`<4-byte IP address>`**: a valid host IP address; a network address is not accepted; use of private IP address is not recommended; class-D IP addresses are rejected
- **`<4-byte AS num>`**: This is a 4-byte AS number; the use of private AS-number is not recommended
- **`<first two bytes of AS num>`**: This is the number represented by the first two bytes of a four-byte AS number. The use of a private AS-number is not recommended.
- **`<last two bytes of AS num>`**: This is the number represented by the last two bytes of a four-byte AS number. The use of a private AS-number is not recommended

Syntax in Set block

```
extended-community  [set | add | delete] "<extended-community-1> <extended-community-2>
…."
extended-community remove
```

Where, the syntax of <extended-community-N> is the following:

```
[rt|soo]:[<2-byte AS num>:<4-byte num> | <4-byte IP Address>:<2-byte num>]
```

The attributes are defined as follows:

- **set**: Replaces the existing extended communities by the new ones as supplied in the policy statement.
- **add**: Adds new extended communities to the existing extended community attribute. If an extended community is already present, then policy will not add a duplicate extended community to the route.
- **delete**: Deletes some of the extended communities from the extended community attribute.
- **remove**: Removes extended community attribute from the route.
- **rt**: route target extended community type
- **soo**: site of origin extended community type
- **<2-byte AS number>**: This is 2-byte AS number. Use of private AS-number is not recommended
- **<4-byte num>**: 4-byte unsigned number
- **<4-byte IP address>**: A valid host IP address. Network address is not accepted. Use of private IP address is not recommended. Class-D IP address will be rejected.

Examples of Extended Communities

The following are examples of valid extended communities:

- rt:10.203.134.56:400
- soo:64500:1600
- rt:64511:2345678
- soo:172.168.45.10:500
- rt:1.15:20000
- soo:65551L:50000

The following are examples of invalid extended communities:

- rt:10.45.87.0:600: Invalid because the IP address is NOT a valid host IP address
- rt:239.1.1.1:400: Invalid because IP address belongs to class-D
- rt:100.200.300.400:200: Invalid because the IP address is invalid
- soo:12345678:500: Invalid because the AS number 12345678 is out of the 2-byte AS number range, 1 to 65535

Extended Community Syntax

Please note the following details with regard to extended community syntax:

- Only rt and soo extended community types are recognized in the policy file.
- The IP address MUST be a valid host address. Network address, Class-D and experimental IP address are not accepted.
- There should not be any blank spaces inside an extended community. For example, rt :100:200 is not a valid extended community because there are spaces between rt and :
- All three parameters of an extended community must be present, otherwise the extended community is rejected.

Extended Community Match Rule in Policy

Regular expressions are not supported for extended communities. In addition, an extended community match statement matches with a route's extended community if at least one of the extended communities in the match statement matches with the route's extended community. There is no need to for all the extended communities in a single match statement to match with the route's extended community.

For example, suppose the policy file is the following:

```
entry one {
if {
extended-community "rt:64500:20000 rt:10.203.134.5:40 soo:64505:50000
soo:192.168.34.1:600";
} then {
permit;
}
}
```

The above community statement will match with all *BGP* routes that have at least one of the following extended communities in their extended community attribute:

- rt:64500:20000
- rt:10.203.134.5:40
- soo:64505:50000
- soo:192.168.34.1:600

Extended Community Set Rule in Policy

A Policy set block can contain several extended community statements. Each set statement is applied to the matching route's extended community attribute in the top down order. That is, the first set is applied to the extended community attribute of the route, the second set is applied to the result of above, and so forth.

For example, assume that a policy is the following:

```
entry two {
if {
nlri 192.168.34.0/24;
} then {
extended-community set "rt:10.45.92.168:300";
extended-community add "rt:10.203.100.200:40 soo:64500:60000";
```

```
extended-community delete "rt:64505:10000 soo:72.192.34.10:70";
permit;
}
}
```

A *BGP* route 192.168.34.128/25 is received with extended community attribute rt:4567:100 soo:192.168.34.128.

When the above policy entry is applied to the route's extended community attribute, the following is true:

- After applying the 1st set (community set "rt:10.45.92.168:300"), the route's community becomes rt:10.45.92.168:300.
- After applying the 2nd set (community add "rt:10.203.100.200:40 soo:64500:60000"), the community becomes rt:10.45.92.168:300 rt:10.203.100.200:40 soo:64500:60000.
- After applying the 3rd set (community delete "rt:64505:10000 soo:72.192.34.10:70"), the community becomes rt:10.45.92.168:300 rt:10.203.100.200:40 soo:64500:60000. Please note that this delete statement has no effect as none of the communities in the delete statement are present in the community attribute.

Extended Communities and BGP Route Aggregation

When *BGP* routes are aggregated with the as-match or as-set CLI option, all the component route's extended community attributes are aggregated and the resulting aggregated extended community attributes are attached to the aggregate network.

Aggregation of several extended community attributes is simply the set union of all the extended communities from all of the aggregated routes.

Route Reflectors

One way to overcome the difficulties of creating a fully meshed AS is to use route reflectors. Route reflectors allow a single router to serve as a central routing point for the AS. All *BGP* speakers in the AS will peer with the route reflector to learn routes.

A cluster is formed by the route reflector and its client routers. Peer routers that are not part of the cluster must be fully meshed according to the rules of BGP.

A BGP cluster, including the route reflector and its clients, is shown in Figure 27.

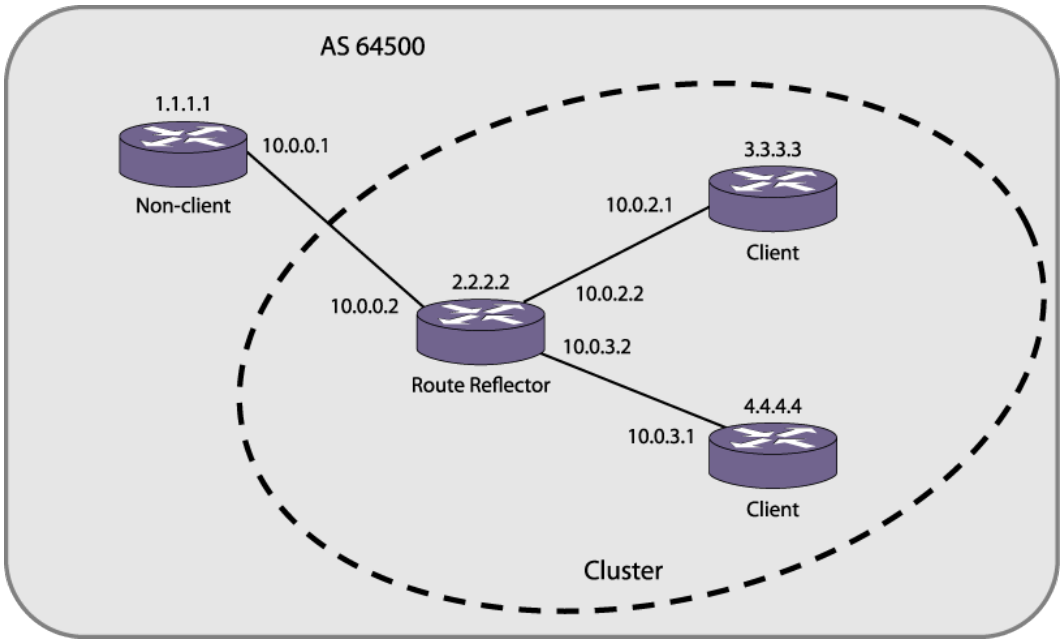


Figure 27: Route Reflectors

This topology minimizes the number of BGP peering sessions required in an AS by using a route reflector.

In this example, although the BGP speakers 3.3.3.3 and 4.4.4.4 do not have a direct BGP peering session between them, these speakers still receive routes from each other indirectly through 2.2.2.2. The router 2.2.2.2 is called a route reflector and is responsible for reflecting routes between its clients. Routes received from the client 3.3.3.3 by the router 2.2.2.2 are reflected to 4.4.4.4 and vice-versa. Routes received from 1.1.1.1 are reflected to all clients.

Multiprotocol BGP

Multiprotocol *BGP* (MBGP), which is also known as BGP4+, is an enhanced BGP that supports more than just IPv4 unicast routes. In this release, MBGP supports:

- IPv4 unicast routing and IPv4 multicast routing on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.
- IPv6 unicast routing and IPv6 multicast routing on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.
- Layer 3 VPN routing on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document..

MBGP support for separate unicast and multicast routing tables allows BGP to have non-congruent topologies for unicast and multicast networks. The BGP multicast address-family routes are used by multicast protocols such as Protocol Independent Multicast (PIM) to build data distribution trees.

Route Confederations

BGP requires networks to use a fully meshed router configuration. This requirement does not scale well, especially when BGP is used as an IGP. One way to reduce the size of a fully meshed AS is to divide the AS into multiple sub-ASs and to group these sub-ASs into a routing confederation. Within the confederation, each sub-AS must be fully meshed. The confederation is advertised to other networks as a single AS.

Figure 28 shows an example of a confederation.

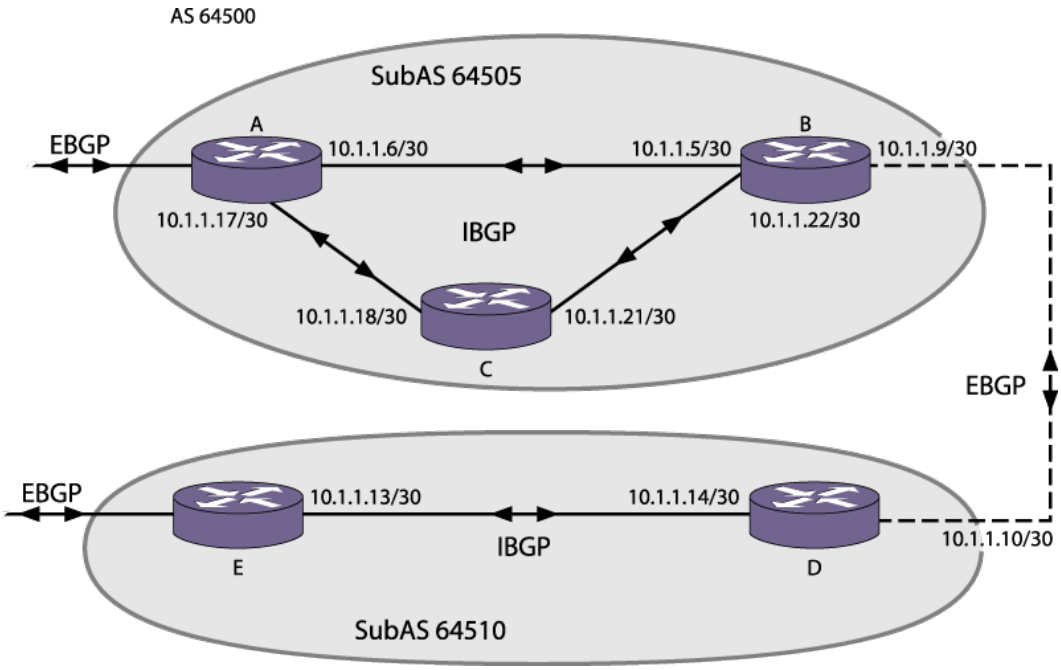


Figure 28: Routing Confederation

In this example, AS 64500 has five BGP speakers. Without a confederation, BGP would require that the routes in AS 64500 be fully meshed. Using the confederation, AS 64500 is split into two sub-ASs: AS 64505 and AS 64510. Each sub-AS is fully meshed, and IBGP is running among its members. EBGP is used between sub-AS 64505 and sub-AS 64510. Router B and router D are EBGP peers. EBGP is also used between the confederation and outside ASs.

Inactive Route Advertisement

BGP inactive routes are defined as those routes that are rated best by BGP and not best in IP routing table. For example, an IGP route to the same destination may be best because it has a higher priority in the IP route table than the BGP best route. The default configuration of the ExtremeXOS software does not advertise BGP inactive routes to BGP neighbors.

The default configuration (no BGP inactive route advertisement) is more consistent with data traffic forwarding. However, when advertisement of inactive BGP routes is enabled, BGP need not depend upon the route manager module to know whether a BGP route is active or not. This actually improves the performance of BGP processing and advertisement.

When BGP inactive route advertising is enabled, inactive BGP routes are considered for BGP route aggregation. When this feature is disabled, inactive BGP routes are ignored while aggregating routes.

Default Route Origination and Advertisement

The default route origination and advertisement feature allows you to originate and advertise a default route to a *BGP* neighbor (or to all neighbors in a peer group) even though no default route exists in the local IP routing table. It also allows you to associate policy rules to conditionally advertise a default route to BGP neighbors.

When default route origination becomes active, the default route is advertised to the specified BGP neighbors, overriding any previously sent default route. If a default route is added to the local IP routing table while default route origination is active, the default route defined by this feature takes precedence over the new regular default route. If default route origination becomes inactive, and a regular default route exists, the regular default route is advertised to BGP neighbors.

When you use a policy with default route origination, the default route is originated only if the local BGP RIB contains a route that matches the policy match conditions. You can use the following match conditions:

- NLRI
- AS-path
- Community
- Origin

You can also use the following policy actions in the policy to set the route attributes:

- AS-path
- Community
- Origin

After a policy is configured for default route origination, BGP must periodically scan the local BGP RIB to make sure that the policy rules evaluate to true for at least one route in local BGP RIB. If the rules evaluate to true, default origination remains active. If the rules evaluate to false, then default origination becomes inactive and the default routes must be withdrawn.

For more information on policy match conditions, actions, and configuration, see Routing Policies.

Using the Loopback Interface

If you are using *BGP* as your IGP, you may decide to advertise the interface as available, regardless of the status of any particular interface. The loopback interface can also be used for EBGP multihop. Using the loopback interface eliminates multiple, unnecessary route changes.

Looped AS_Path Attribute

When an EBGP speaker receives a route from its neighbor, it must validate the AS_Path attribute to ensure that there is no loop in the AS_Path. When an EBGP speaker finds it's own AS-number in the received EBGP route's AS_Path attribute, it is considered as a Looped AS Path and by default, the associated EBGP routes are silently discarded.

There are certain cases (such as in a spoke and hub topology) where it becomes necessary to accept and process EBGP routes with a looped AS_Path attribute.

This feature enables you to control the processing of EBGP routes with a looped AS_Path attribute. You can do the following:

- Allow the route with a looped AS_Path to be accepted
- Allow the route with a looped AS_Path to be accepted only if the number of occurrences of its own AS-Number is less than or equal to N, where N is an user-configurable unsigned integer configured
- Silently discard the route with looped AS_Path

Note
A looped AS path is always allowed for IBGP, irrespective of the *BGP* configuration.

BGP Peer Groups

You can use *BGP* peer groups to group together up to 512 BGP neighbors. All neighbors within the peer group inherit the parameters of the BGP peer group. The following mandatory parameters are shared by all neighbors in a peer group:

- remote AS
- source-interface
- route-policy
- send-community
- next-hop-self

Each BGP peer group is assigned a unique name when it is created, and each peer supports either IPv4 or IPv6 address families (but not both).

When the first peer is added to a BGP peer group, the peer group adopts the IPv4 or IPv6 address family of the assigned peer. After the first peer is assigned, the peer group does not support any configuration options or capabilities for the IP address family not selected. For example, if an IPv6 peer is the first peer added to a peer group, the peer group no longer supports IPv4 configuration options or capabilities.

Changes made to the parameters of a peer group are applied to all neighbors in the peer group.

Modifying the following parameters will automatically disable and enable the neighbors before changes take effect:

- remote-as
- timer

- source-interface
- soft-in-reset
- password

Bidirectional Forwarding Detection (BFD) for BGP

Bidirectional Forwarding Detection (BFD) protection of *BGP* peering sessions allows for the rapid detection of link failures such that peering sessions can be taken out of the "established" state within fractions of a second. This allows the protocol to select an alternate path (if available) to a destination immediately after the link failure, rather than waiting until the BGP hold timer expires (180 seconds by default).

This feature applies to both IPv4 and IPv6 peering sessions. Both IPv6 global and link local peering sessions are supported.

Multi-hop BFD is introduced in ExtremeXOS 30.5. This enhancement allows BFD support for non-directly connected EBGP sessions, as well as IBGP sessions. Multi-hop BFD is supported in hardware assist BFD also. There are no new BGP configuration or show commands associated with multi-hop BFD. However, the `show bfd session {ipv4 | ipv6} {ipaddress } detail {vr [vrname | all]}` command shows "Session Type" as "Multi Hop" for multi-hop BFD and "Single Hop" for single-hop BFD.

Limitations

- The BFD setting can be applied on a per-peer basis, but the ability to set BFD on a peer-group or address-family basis is not currently supported.
- The BGP peer must be in the disabled admin state to modify its BFD setting.
- While BFD can be enabled on any BGP peering session, protection is only provided for directly connected EBGP peering sessions.

BGP Route Flap Dampening

Route flap dampening is a *BGP* feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when it repeatedly alternates between being available and being unavailable. Without route flap dampening, each transition of a route produces a withdrawal or advertisement message, which is propagated throughout the network. Route flapping can generate large numbers of messages, which can impact network bandwidth and availability.

The route flap dampening feature minimizes the flapping problem by halting route advertising and withdrawal messages for the affected route for a period of time. To support route flap dampening, the ExtremeXOS software employs a combination of two techniques. The first technique uses fixed timers to reduce the frequency of route advertisement as specified in RFC 4271. The other technique uses route flap damping algorithms. The software uses a combination of both techniques as specified in RFC 2439.

The fixed timers technique blocks all updates for the flapping route for a period defined by the internal MinRouteAdvertisementInterval (MRAI) timer (which is not

configurable). For IBGP routes, this timer is set for 5 seconds. For EBGP routes, this timer is set to 30 seconds. The MRAI timer check is independent of the dampening configuration and is used to limit the frequency of route advertisements.

The route flap dampening algorithm uses configurable timers to manage route flap dampening. Suppose that the route to network 172.25.0.0 flaps. The router (in which route dampening is enabled) responds by doing the following:

- Assigns route 172.25.0.0 a penalty of 1000 and moves it to a history state in which the penalty value is monitored.
- Increments the penalty value by 1000 for each additional route flap.
- Accumulates penalties and compares them to the suppression limit, which is set to 2000 by default.

If the suppression limit is exceeded when the MRAI timer expires, the route is not advertised to neighbors.

A route remains suppressed or dampened until one of the following events occurs:

- The suppression limit is not met when the MRAI timer expires.
- The penalty placed on network 172.25.0.0 is decayed below the reuse limit.
- The maximum suppression timer expires.

The penalty is decayed by reducing the penalty value by one-half at the end of a configurable time period, called the half-life. Routes that flap many times may reach a maximum penalty level, or ceiling, after which no additional penalty is added. The ceiling value is not directly configurable, but the configuration parameter used in practice is the maximum route suppression time. No matter how often a route has flapped, after it stops flapping, it is advertised after the maximum route suppression time.

BGP Route Selection

BGP selects routes based on the following precedence (from highest to lowest):

- Next-hop must be reachable.
- higher weight
- higher local preference
- shortest length (shortest AS path)
- lowest origin code
- lowest Multi Exit Discriminator (MED)
- route from external peer
- lowest cost to next hop
- lowest routerID
- lowest PeerID

Private AS Number Removal from Route Updates

Private AS numbers are AS numbers in the range 64512 through 65534. You can remove private AS numbers from the AS path attribute in updates that are sent to external *BGP* (EBGP) neighbors. Possible reasons for using private AS numbers include:

- The remote AS does not have officially allocated AS numbers.
- You want to conserve AS numbers if you are multihomed to the local AS.

Private AS numbers should not be advertised on the Internet. Private AS numbers can be used only locally within an administrative domain. Therefore, when routes are advertised out to the Internet, the routes can be stripped out from the AS paths of the advertised routes using this feature.

Route Redistribution

More than one routing protocol can be enabled simultaneously on the switch.

Route redistribution allows the switch to exchange routes, including static routes, between the routing protocols. Figure 29 is an example of route redistribution between an OSPF AS and a RIP AS.

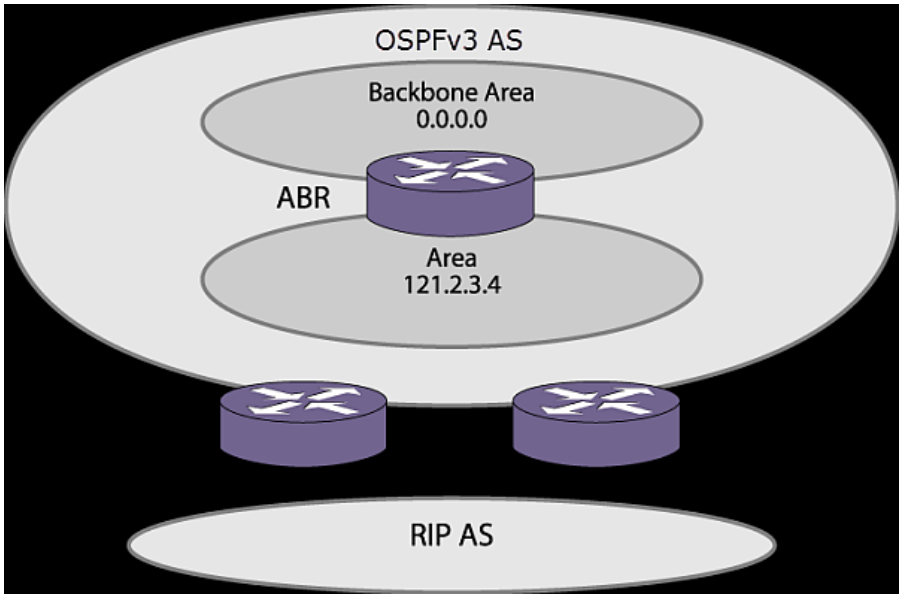


Figure 29: OSPF Route Redistribution

BGP ECMP

The *BGP ECMP (Equal Cost Multi Paths)* feature supports load sharing by creating a multipath to a destination. This multipath contains multiple routes that are determined to have an equal cost because the following parameters are the same for each route:

- Weight
- Local preference (for IBGP multipaths)
- AS path (entire attribute, not just the length)

- Origin code
- Multi Exit Discriminator (MED)
- IGP distance to the next hop
- Source session (EBGP or IBGP)

Note
ECMP does not install an additional path if the next hop is the same as that of the best path. All paths within a multipath must have a unique next hop value.

BGP ECMP does not affect the best path selection. For example, the router continues to designate one of the paths as the best path and advertise this best path to its neighbors. EBGP paths are preferred to IBGP paths.

The BGP ECMP feature allows you to define the maximum number of equal cost paths (up to 64) in a multipath. A multipath for an IBGP destination is called an IBGP multipath, and the multipath for an EBGP destination is called an EBGP multipath.

If there are more equal cost paths for a destination than the configured maximum, the BGP identifier for the advertising BGP speaker is used to establish a path priority. The lower BGP identifier values have priority over the higher values. For example, if the configuration supports 4 paths in a multipath, only the four paths with the lowest BGP identifier values become part of the multipath.

BGP Mulitipath-Relax

This feature overrides the default protocol behavior specified in *RFC-4271* related to route selection tie breakers and the definition of equal cost. In particular, routes with the same AS-path length, but differing AS numbers in the path are not considered equal cost by default. However, with multipath-relax enabled, routes with the same AS-path length can have differing AS number values in the AS-path and still be considered equal cost.

By default, BGP considers two routes as equal cost if the costs are equal through step (e) of the tie breaker procedure specified in Section 9.1.2.2 of *RFC-4271*. Step (a) of this procedure specifies that the AS-path length must be the same and the AS values within the path must also be the same. When the multipath-relax feature is enabled, the restriction that the AS values themselves must be the same is “relaxed”. So two routes with a different AS values in the path can be consider equal cost provided that the AS-path length is equal and the remaining criteria specified in the tie-breaker procedure [through step (e)] are the also equal. Note that multipath support must be also be enabled on the device both in BGP and RTMGR for this feature to have any effect.

BGP Static Network

ExtremeXOS *BGP* allows users to add static networks in BGP, which will be redistributed (advertised) into the BGP domain if there is a corresponding active route in the IP routing table. Users can associate a policy with the static BGP network to change or to set the route attributes before the route is advertised to the BGP neighbors.

Graceful BGP Restart

The graceful *BGP* restart feature enables the switch BGP process to restart without disrupting traffic forwarding. This feature also enables the BGP process to support the graceful BGP restart of a peer router. Without graceful restart, BGP routes within the restarting router are flushed, and BGP routes through the restarting router are flushed from BGP peers. A non-graceful restart interrupts traffic for the time it takes for BGP to restart and re-establish routes, and expends additional resources during the reconvergence.

Graceful Restart in the Restarting Switch

During session startup, *BGP* peers indicate whether they have the graceful restart capability. When BGP restarts, the restarting router indicates during the new session startup that it is restarting and provides a restart-time value, which defines how long the receiving router will wait for the restart to complete.

In the restarting router, a graceful BGP restart preserves the BGP routes in the routing table, which allows the router to continue to use those routes until one of the following events occurs:

- The BGP restart is complete.
- The restart update delay timer expires.

During the restart, all pre-existing routes are marked stale. BGP receives new routes during the restart, but the Routing Information Base (RIB) is not updated and advertised until the restart is complete. An End of RIB message is sent when the restart is complete if the graceful restart feature is enabled.

An update-delay timer value determines how long the restarting switch will wait before updating the stale routes. If this timer expires before the restarting switch receives updates from the receiving routers, all stale routes are deleted. If the receiving routers provide updates before this timer expires, the time-stamps for any matching entries in the local RIB are preserved.

After the new BGP session is established, the new session uses the capabilities established with that session, which includes any updates to the graceful restart capability or timers.

Graceful Restart on the Receiving Switch

During session startup, *BGP* peers indicate whether they have the graceful restart capability. When BGP restarts, the receiving router retains routes received from the restarting router and marks those routes stale. The receiving router continues to advertise the restarting router as if it was fully functional until one of the following events occurs:

- The restarting router sends the EOR marker, indicating the end of a routing update and the end of the graceful restart.
- The restart timer defined by the restarting router expires.
- The stale-route-time timer defined on the receiving router expires.

If the receiving router receives RIB updates and the EOR marker before the timers expire, it updates the local RIB and deletes any stale entries. If either of the timers on the receiving router expires before the receiving switch receives the EOR marker from the restarting switch, all stale routes are deleted.

Planned and Unplanned Restarts

Two types of graceful restarts are defined: planned and unplanned.

A planned restart occurs when the software module for *BGP* is upgraded, or the router operator decides to restart the BGP control function. An unplanned restart occurs when there is some kind of system failure that causes a remote reboot or a crash of BGP, or when a failover occurs. You can decide to configure a router to enter graceful restart for only planned restarts, for only unplanned restarts, or for both. Also, you can decide to configure a router to be a receiver only, and not to do graceful restarts itself.

Cease Subcodes

BGP uses the cease subcode in notification message to convey the reason for terminating the session. The cease subcodes currently supported are given in Table 31.

Table 31: Supported Cease Subcodes

Subcode	Description	Supported?
1	Maximum Number of Prefixes Reached	Yes
2	Administrative Shutdown	Yes
3	Peer De-configured	Yes
4	Administrative Reset	No
5	Connection Rejected	Yes
6	Other Configuration Change	No
7	Connection Collision Resolution	Yes
8	Out of Resources	No

Maximum Number of Prefixes Reached

This cease subcode is sent when the number of prefixes from a *BGP* neighbor exceeds the pre-configured limit. The notification message contains additional data to indicate the maximum prefix limit configured for the neighbor.

Administrative Shutdown

This cease notification subcode is sent to a *BGP* neighbor in following two situations:

- BGP neighbor is disabled
- BGP protocol is globally disabled. All BGP neighbors that were in the established state send cease notifications with this subcode

Peer De-configured

This cease notification subcode is sent when *BGP* neighbor is deleted.

Other Configuration Change

This cease notification subcode is sent when the following configuration entities change:

- *BGP* neighbor is added to a peer group
- BGP neighbor is configured as a route-reflector client
- BGP neighbor is part of a peer group and the following configuration elements of the peer group are changed:
  - Password
  - Remote-as
  - Hold-time, keepalive-time
  - Source interface
  - Soft-in-reset

Connection Collision Resolution

This cease notification subcode is sent when there is a *BGP* connection collision.

Fast External Fallover

BGP fast external fallover uses the BGP protocol to converge quickly in the event of a link failure that connects it to an EBGP neighbor.

When BGP fast external fallover is enabled, the directly-connected EBGP neighbor session is immediately reset when the connecting link goes down and routes are learned over the BGP session. Failover is not initiated if routes are not learned over the BGP session.

If BGP fast external fallover is disabled, BGP waits until the default hold timer expires (3 keepalives) to reset the neighboring session. In addition, BGP may tear down the session somewhat earlier than hold timer expiry if BGP detects that the TCP session and it's directly connected link is broken (BGP detects this while sending or receiving data from the TCP socket).

Capability Negotiation

BGP supports the negotiation of the following capabilities between BGP peers:

- IPv4 unicast address family
- IPv4 multicast address family
- IPv6 unicast address family
- IPv6 multicast address family
- Route-refresh (code = 64 and Cisco-style code = 128)
- 4-byte-AS (code = 65)

BGP brings up peering with the minimal common capability for the both sides. For example, if a local router with both unicast and multicast capabilities peers with a remote router with unicast capability, the local router establishes the connection with unicast-only capability.

When there are no common capabilities, BGP sends an Unsupported Capability error and resets the connection. A manual intervention and configuration change might be required in order to establish a BGP peering session in this particular case.

Note
ExtremeXOS version 12.0 and earlier does not negotiate address families. By default, ExtremeXOS version 12.1 advertises MBGP options and is rejected by switches running previous versions, which can delay the establishment of a BGP session. We recommend that you disable the capability advertisement feature of MBGP while peering with switches running previous versions of ExtremeXOS for faster neighbor session establishment.

IPv4 Capability Negotiation

For IPv4 peers, *BGP* supports the following capabilities by default:

- IPv4 unicast address family
- IPv4 multicast address family
- Route-refresh (code = 64 and Cisco-style code = 128)
- 4-byte-AS (code = 65)

By default, BGP sends those capabilities in its OPEN message. In addition, BGP supports graceful restart.

When BGP receives a notification 2/4 (Unsupported optional parameters) in response to an OPEN, it assumes that the peer does not support capability negotiation and MBGP and sends an OPEN message without any capability.

If the peer speaker sends no capabilities, but the local speaker is configured for the IPv4 unicast capability, the assumption is that the peer speaker is operating in legacy mode, and the session defaults to the exchange of IPv4 unicast NLRIs (a non MBGP session).

If the local speaker is configured explicitly with the IPv4 unicast family disabled, it cannot peer with legacy peers, and it will send the Optional Attribute Error whenever it receives an update packet. Because the IPv4 address family is enabled for Extreme Networks switches by default, it is recommended that you explicitly disable this address family when you desire non-standard behavior.

IPv6 Capability Negotiation

For IPv6 peers, the route refresh capability is enabled by default, and no address family is enabled by default. You must enable a common set of capabilities on the local and neighbor peers before peering can be established. For IPv6 capability negotiation, IPv6 peering is set to idle if no common address families are negotiated. IPv6 *BGP* peering supports only the IPv6 unicast and IPv6 multicast address families; IPv4 address families are not supported.

Route Refresh

Route Refresh helps minimize the memory footprint of *BGP* by not storing the original BGP route path attributes from a neighbor that advertises route refresh capability in an OPEN message. Whenever you execute the command `configure bgp neighbor [remoteAddr | all] {addressfamily [ipv4-unicast | ipv4-multicast]} soft-reset in`, BGP sends a route refresh message to its peer if that peer had advertised route refresh capability in its OPEN message. In response to the route refresh message, the neighbor sends its entire RIB-OUT database for the requested address family. This helps reapply the inbound neighbor policy, if there are any changes.

Configuring BGP

Configuration Overview

The following procedure configures a basic *BGP* topology:

1. Configure the interfaces that will connect to BGP neighbors. For each interface, do the following:
   a. Create a *VLAN (Virtual LAN)*.
   b. Assign one or more ports to the VLAN.
   c. Configure a VLAN IP address.
   d. Enable IP forwarding on the VLAN.

   For more information on configuring VLANs, see VLANs on page 2213
2. Configure the BGP router ID using the following command:

   configure bgp routerid router identifier
3. Configure the AS number to which the router should belong using the following command:

   `configure bgp AS-number number`
4. Configure the BGP neighbor retry timer using the following command:

   `configure bgp neighbor [all | remoteaddr] connect-retry seconds`
5. To configure the BGP retry timer to a specified peer group, use the following command:

   `configure bgp peer-group peer-group-name connect-retry seconds`
6. To add one or more IBGP neighbors, use the following command and specify the AS number to which the router belongs:

   `create bgp neighbor remoteaddr remote-AS-number as-number {multi-hop}`
7. To add one or more EBGP neighbors, use the following command and specify the AS number of the remote AS (which is different from the AS to which the router belongs):

   `create bgp neighbor remoteaddr remote-AS-number as-number {multi-hop}`
8. If you want to simultaneously configure BGP options for multiple neighbors, create and configure peer groups as described in Configuring BGP Peer Groups.

9. If the BGP network will support IPv4 traffic, you can skip this step. You can also skip this step, if for an IPv6 BGP neighbor, you only want ipv6-unicast, which is automatically enabled when you create the peer. If the BGP network will support any other address family, you must enable support for that address family on BGP neighbors with either of the following commands:

   ```
   enable bgp neighbor [all |remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]
   ```

   ```
   enable bgp peer-group peer-group-name capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh]
   ```

10. To configure additional BGP neighbor options, see Configuring BGP Neighbors.
11. For instructions on configuring additional BGP features, see the list under Configuring BGP.
12. Enable BGP neighbors using the following command:

    ```
    enable bgp neighbor [remoteaddr | all]
    ```

13. Enable BGP using the following command:

    ```
    enable bgp
    ```

For instructions on displaying BGP information, see Displaying BGP Information.

Configuring BGP Router Settings

Configure the BGP Router ID

- To configure the *BGP* router ID, use the following command:

  ```
  configure bgp routerid router-identifier
  ```

Configure the AS Number

- To configure the AS number for the switch., use the following command:

  ```
  configure bgp AS-number number
  ```

Configure the AS Number and Community Display Formats

- To configure the AS number display format, use the following commands:

  ```
  configure bgp as-display-format [asdot | asplain]
  ```

  ```
  enable bgp community format AS-number : number
  ```

  ```
  disable bgp community format AS-number : number
  ```

Configure the BGP Local Preference

- To configure the *BGP* local preference, use the following command:

  ```
  configure bgp local-preference number
  ```

Configure the BGP MED

- To configure the *BGP* MED, use the following command:

  ```
  configure bgp med [none | bgp_med]
  ```

```
enable bgp always-compare-med

disable bgp always-compare-med
```

Configure BGP ECMP

- To enable or disable *BGP ECMP*, enter the following command:
  `configure bgp maximum-paths max-paths`

  The *max-paths* setting applies to BGP on the current VR. Specify more than 1 path to enable BGP ECMP and define the maximum number of paths for IBGP and EBGP multipaths. Specify 1 path to disable ECMP. To display BGP ECMP configuration information, use the `show bgp` command.

Configure BGP Multipath-Relax

- To enable BGP multipath-relax, use the following command:

  `enable bgp` **`multipath-relax`**
- To disable BGP multipath-relax, use the following command:

  `disable bgp` **`multipath-relax`**
- To display the status or BGP multipath-relax, use the following command:

  `show bgp`

Configure Graceful BGP Restart

- To configure a router to perform graceful *BGP* restart, use the following command:
  `configure bgp restart [`**`none`** | **`planned`** | **`unplanned`** | **`both`** | **`aware-only`**`]`
- The address families participating in graceful restart are configured using the following command:
  `configure bgp restart [`**`add`** | **`delete`**`] address-family [`**`ipv4-unicast`** | **`ipv4-multicast`** | **`ipv6-unicast`** | **`ipv6-multicast`**`]`
- There are three timers that can be configured with the following commands:
  `configure bgp restart restart-time` *seconds*
  `configure bgp restart stale-route-time` *seconds*
  `configure bgp restart update-delay` *seconds*
- You can use the following commands to verify the BGP graceful restart configuration:
  `show bgp`
  `show bgp {`**`neighbor`**`}` *remoteaddr* `{`**`address-family`** `[`**`ipv4-unicast`** | **`ipv4-multicast`** |**`ipv6-unicast`** | **`ipv6-multicast`** | **`ipv4-vxlan`** | `{`**`l2vpn-evpn`** `[`**`inclusive-multicast`** | **`mac-ip`** | **`auto-discovery`** | **`esi`** | **`ip-prefix`**`]}]}` `[`**`accepted-routes`** | **`received-routes`** | **`rejected-routes`** | **`transmitted-routes`**`] {`**`detail`**`} [`**`all`** | **`as-path`** *path-expression* | **`community`** `[`**`no-advertise`** | **`no-export`** | **`no-export-subconfed`** | **`number`** *community_number* | *autonomous-system-id* **`:`** *bgp-community*`]` | **`network`** `[`**`any/`***netMaskLen* | **`networkPrefixFilter`**`] {`**`exact`**`}]`

Configuring Fast External Fallover

The fast external fallover module consists of two commands; one for enabling fallover (`enable bgp fast-external-fallover`) and one for disabling fallover (`disable bgp fast-external-fallover`). Fast external fallover is disabled by default.

These commands apply to all directly-connected external BGP neighbors.

Configuring BGP Neighbors

Create and Delete BGP Neighbors

- To create or delete *BGP* neighbors, , use the following commands:

  `create bgp neighbor` *`remoteaddr`* `remote-AS-number` *`as-number`* {**`multi-hop`**}

  `create bgp neighbor` *`remoteaddr`* `peer-group` *`peer-group-name`* {**`multi-hop`**}

  `delete bgp neighbor [`*`remoteaddr`* | **`all`**`]`

Configure a Description for a Neighbor

- To configure a description for one or all *BGP* neighbors, use the following command:

  `configure bgp neighbor [`**`all`** | *`remoteaddr`*`] description {`*`description`*`}`

Configure a Password for Neighbor Communications

- To configure a password to use for communications with *BGP* neighbors, use the following command:

  `configure bgp neighbor [`**`all`** | *`remoteaddr`*`] password [`**`none`** | {**`encrypted`**} *`tcpPassword`*]

- To configure a password for the neighbors in a peer group, use the following command:

  `configure bgp peer-group` *`peer-group-name`* **`password`** [**`none`** | {**`encrypted`**} *`tcpPassword`*]

Configure the Supported Address Families and Route Refresh

- All *BGP* negotiated capabilities (except for the 4-Byte-AS capability) can be enabled and disabled using the following commands:

  `enable bgp neighbor [`**`all`** |*`remoteaddr`*`]` **`capability`** [**`ipv4-unicast`** | **`ipv4-multicast`** | **`ipv6-unicast`** | **`ipv6-multicast`** | **`vpnv4`** | **`route-refresh`** | **`ipv4-vxlan`** | **`l2vpn-evpn`**]

  `disable bgp neighbor [`**`all`** | *`remoteaddr`*`]` **`capability`** [**`ipv4-unicast`** | **`ipv4-multicast`** | **`ipv6-unicast`** | **`ipv6-multicast`** | **`vpnv4`** | **`route-refresh`** | **`ipv4-vxlan`** | **`l2vpn-evpn`**]

- To configure the capabilities for a peer group, use the following command:

  `enable bgp peer-group` *`peer-group-name`* `capability [`**`ipv4-unicast`** | **`ipv4-multicast`** | **`ipv6-unicast`** | **`ipv6-multicast`** | **`vpnv4`** | **`route-refresh`**]

```
disable bgp peer-group peer-group-name capability [ipv4-unicast |
ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-
refresh]
```

Note
ExtremeXOS allows the sending and receiving of IPV6 routes over IPV4 peering sessions, and IPV4 routes over IPV6 peering sessions.

Configure Timers for BGP Neighbor Communications

- Configure the timers that apply to communications with a neighbor.
  `configure bgp neighbor [remoteaddr | all] timer keep-alive keepalive hold-time holdtime`
- Configure the timers that apply to the neighbors in a peer group.
  `configure bgp peer-group peer-group-name timer keep-alive seconds hold-time seconds`

Configure the Neighbor Shutdown Priority

- To configure the neighbor shutdown priority, use the following command:
  `configure bgp neighbor [all | remoteaddr] shutdown-priority number`

Setting an Alternative Local Autonomous System (AS)

To allow the local router to accept peering sessions intended for the specified alternate local autonomous system (AS), use the following command:

```
configure bgp neighbor [all | remoteaddr] alternate-local-as asNumber
```

Configuring Route Acceptance

Assign a Weight Value to Routes Learned from a Neighbor

- To configure the weight value that applies to routes learned from a neighbor, use the following command:
  `configure bgp neighbor [remoteaddr | all] weight weight`
- To configure the weight value that applies to routes learned from the neighbors in a peer group, use the following command:
  `configure bgp peer-group peer-group-name weight weight`

Configure the Maximum Number of Prefixes

- To configure the maximum number of prefixes to accept from a *BGP* neighbor, use the following command:
  `configure bgp neighbor [remoteaddr | all] {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} maximum-prefix number {{threshold percent} {teardown {holddown-interval seconds}} {send-traps}`
- To configure the maximum number of prefixes to accept from the neighbors in a peer group, use the following command:
  `configure bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} maximum-`

```
prefix number {{threshold percent} {teardown {holddown-interval
seconds}} {send-traps}
```

Configure Acceptance of Looped BGP Routes from Neighbors

- To enable and disable the acceptance of looped routes from one or all neighbors, use the following commands:

  ```
  configure bgp neighbor [all | remoteaddr] {address-family [ipv4-
  unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]}
  allowas-in {max-as-occurrence as-count}
  ```

  ```
  configure bgp neighbor [all | remoteaddr] {address-family [ipv4-
  unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]}
  dont-allowas-in
  ```

- To enable and disable the acceptance of looped routes from the neighbors in a peer group, use the following commands:

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast
  | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} allowas-in
  {max-as-occurrence as-count}
  ```

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast
  | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} dont-
  allowas-in
  ```

Configuring Route Origination

Configure the Source Interface Address

- To configure the source interface address to use for communications with a neighbor, use the following command:

  ```
  configure bgp neighbor [remoteaddr | all] source-interface [any |
  ipaddress ipAddr]
  ```

- To configure the source interface address to use for communications with neighbors in a peer group, use the following command:

  ```
  configure bgp peer-group peer-group-name source-interface [any |
  ipaddress ipAddr]
  ```

Enable and Disable Default Route Origination

- To enable or disable *BGP* default route origination and advertisement for BGP neighbors, use the following commands:

  ```
  enable bgp [{neighbor} remoteaddr | neighbor all] {address-family
  [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]}
  originate-default {policy policy-name}
  ```

  ```
  disable bgp [{neighbor} remoteaddr | neighbor all] {address-family
  [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]}
  originate-default
  ```

- To enable or disable BGP default route origination and advertisement for a peer group, use the following commands:

  ```
  enable bgp {peer-group} peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]} originate-default {policy policy-name}
  ```

  ```
  disable bgp {peer-group} peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]} originate-default
  ```

Configure Inactive Route Advertisement

This command applies to the specified address family for all neighbors.

- To enable or disable *BGP* inactive route advertising, use the following commands:

  ```
  enable bgp {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]} advertise-inactive-route
  ```

  ```
  disable bgp {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]} advertise-inactive-route
  ```

Configure the Originating Next Hop Address for Outgoing Updates

- To configure outgoing updates to the specified neighbors to specify the address of the *BGP* connection originating the update as the next hop address, use the following command:

  ```
  configure bgp neighbor [remoteaddr | all] {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} [next-hop-self | no-next-hop-self]
  ```

- To make a configuration change for the neighbors in a peer group, use the following command:

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} [next-hop-self | no-next-hop-self]
  ```

Include or Exclude the Community Path Attribute

- To configure neighbor communications to include or exclude the community path attribute, use the following command:

  ```
  configure bgp neighbor [remoteaddr | all] {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} [send-community | dont-send-community] {both | extended | standard}
  ```

- To make a configuration change for the neighbors in a peer group, use the following command:

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} [send-community | dont-send-community] {both | extended | standard}
  ```

Remove Private AS Numbers from Route Updates

- To configure private AS numbers to be removed from updates for neighbors or peer groups, use the following commands:

  ```
  enable bgp neighbor [remoteaddr | all] remove-private-AS-numbers
  ```

```
enable bgp peer-group peer-group-name remove-private-AS-numbers
```

- To disable this feature, use the following commands:

  ```
  disable bgp neighbor [remoteaddr | all] remove-private-AS-numbers
  disable bgp peer-group peer-group-name remove-private-AS-numbers
  ```

Configure a Route Map Filter

- To configure a route map filter for one or all *BGP* neighbors, use the following command:

  ```
  configure bgp neighbor [remoteaddr | all] {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} route-policy [in | out] [none | policy]
  ```

- To configure a route map filter for the neighbors in a peer group, use the following command:

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} route-policy [in |out] [none | policy]
  ```

Enable and Disable the Soft Input Reset Feature for a Neighbor

- To enable or disable the soft input reset feature for a neighbor, use the following commands:

  ```
  enable bgp neighbor [all |remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]
  disable bgp neighbor [all | remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]
  ```

- Enable or disable the soft input reset feature for a peer group.

  ```
  enable bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast |ipv6-unicast |ipv6-multicast |vpnv4]}soft-in-reset
  disable bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast |ipv6-unicast | ipv6-multicast]} soft-in-reset
  ```

Configure Route Flap Dampening

You can supply the dampening configuration parameters directly through a command line interface (CLI) command, or use the command to associate a policy that contains the desired parameters.

- To enable route flap dampening for neighbors or a peer group, use the following commands:

  ```
  configure bgp neighbor [all | remoteaddr] {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} dampening {{half-life half-life-minutes {reuse-limit reuse-limit-number suppress-limit suppress-limit-number max-suppress max-suppress-minutes} | policy-filter [policy-name | none]}
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]}
  ```

```
dampening {{half-life half-life-minutes {reuse-limit reuse-limit-
number suppress-limit suppress-limit-number max-suppress max-suppress-
minutes} | policy-filter [policy-name | none]}
```

- To disable route flap dampening for a BGP neighbor or peer group, use the following commands:

  ```
  configure bgp neighbor [remoteaddr | all] {address-family [ipv4-
  unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} no-
  dampening
  ```

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast
  | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} no-
  dampening
  ```

  Note
  When you disable dampening, all the configured dampening parameters are deleted.

Configuring Bidirectional Forwarding Detection (BFD) on BGP Neighbor

To enable or disable BFD on BGP neighbor(s):

```
configure bgp {neighbor [all | remoteaddr ]} {bfd [on | off]}
```

Configure Maximum AS Path Length

- To configure maximum-as-path-length in the switch, use the following command:

  ```
  configure bgp maximum-as-path-length [max-as-path | none]
  ```

  Note
  This configuration is independent of peers, and it is applied to entire *BGP*.

Configuring BGP Peer Groups

Create or Delete a BGP Peer Group

To create or delete peer groups, use the following commands:

- `create bgp peer-group peer-group-name`
- `delete bgp peer-group peer-group-name`

Note
No capabilities are enabled at time of peer-group creation. If an IPv4 peer is the first peer added to the peer-group, the IPv4-unicast and IPv4-multicast capabilities are enabled by default. If the first peer assigned to a peer-group is an IPv6 peer, no capabilities are enabled.

Add Neighbors to a BGP Peer Group

- To create a new neighbor and add it to a *BGP* peer group, use the following command:

  ```
  create bgp neighbor remoteaddr peer-group peer-group-name {multi-hop}
  ```

The new neighbor is created as part of the peer group and inherits all of the existing parameters of the peer group. The peer group must have remote AS configured.

- To add an existing neighbor to a peer group, use the following command:

  `configure bgp neighbor [**all** | *remoteaddr*] **peer-group** [*peer-group-name* | **none**] {**acquire-all**}`

  If you do not specify the acquire-all option, only the mandatory parameters are inherited from the peer group. If you specify the acquire-all option, all of the parameters of the peer group are inherited. This command disables the neighbor before adding it to the peer group.

  To remove a neighbor from a peer group, use the peer-group none option.

  When you remove a neighbor from a peer group, the neighbor retains the parameter settings of the group. The parameter values are not reset to those the neighbor had before it inherited the peer group values.

Configure a Remote AS Number for a Peer Group

- To configure a remote AS number for a peer group, use the following command:

  `configure bgp peer-group *peer-group-name* remote-AS-number *number*`

Create and Delete BGP Static Networks

- To create a static *BGP* network, use the following command:

  `configure bgp add network {**address-family** [**ipv4-unicast** | **ipv4-multicast** | **ipv6-unicast** | **ipv6-multicast**]} *ipaddress/masklength* {**network-policy** *policy*}`

  Note
  This command adds the route to BGP only if the route is present in the routing table.

- To delete a static BGP network, use the following command:

  `configure bgp delete network {**address-family** [**ipv4-unicast** | **ipv4-multicast** | **ipv6-unicast** | **ipv6-multicast**]} [**all** | *ipaddress/mask length*]`

Import Routes from Other Protocols to BGP

Before importing routes from another protocol to *BGP*, you must first configure both protocols and then verify the independent operation of each. When you import routes from another protocol, the imported routes are limited to those for the respective address family and protocol. For example, you can import IPv4 unicast routes from *RIP (Routing Information Protocol)* to BGP, but you cannot import IPv6 unicast routes from RIP to BGP. IPv6 routes can be imported only from IPv6 routing protocols; and IPv4 routes can be imported only from IPv4 routing protocols.

You can use route maps to associate BGP attributes including Community, NextHop, MED, Origin, and Local Preference with the routes. Route maps can also be used to filter out exported routes.

For instructions on importing routes from another protocol to BGP, refer to the chapter for the protocol from which you want to import routes.

- To configure an import policy to apply to imported routes, use the following command:

  ```
  configure bgp import-policy [policy-name | none]
  ```

  For Layer 3 VPNs, a local PE stores the routes received from a remote PE in the VPN-VRF RIB.
- To enable or disable export of these routes as IPv4 unicast routes to the CE, use the following command:

  ```
  enable bgp export remote-vpn {export-policy} policy-name} {address-family [ipv4-unicast | vpnv4]}
  disable bgp export remote-vpn {address-family [ipv4-unicast | vpnv4]}
  ```

Export BGP Routes to other Protocols

Before exporting routes from *BGP* to another protocol, you must first configure both protocols and then verify the independent operation of each. When you export routes from BGP, the exported routes are limited to those for the respective address family and protocol. For example, you can export IPv4 unicast routes from BGP to *RIP*, but you cannot export IPv6 unicast routes from BGP to RIP. IPv6 routes can be exported only from BGP to IPv6 routing protocols; and IPv4 routes can be exported only to IPv4 routing protocols.

You can use route maps to associate BGP attributes including Community, NextHop, MED, Origin, and Local Preference with the routes. Route maps can also be used to filter out exported routes.

- To enable or disable the export of routes into BGP from other routing sources like ospf and rip, use the following commands:

  ```
  enable bgp export [blackhole | direct | isis | isis-level-1 | isis-level-1-external | isis-level-2 | isis-level-2-external | ospf | ospf-extern1 | ospf-extern2 | ospf-inter | ospf-intra | ospfv3 | ospfv3-extern1 | ospfv3-extern2 | ospfv3-inter | ospfv3-intra | rip | ripng | static {address-family [{ipv4-unicast |ipv4-multicast | ipv6-unicast | ipv6-multicast]} {export-policy policy-name}
  disable bgp export [blackhole | direct | isis | isis-level-1 | isis-level-1-external | isis-level-2 | isis-level-2-external | ospf | ospf-extern1 | ospf-extern2 | ospf-inter | ospf-intra | ospfv3 | ospfv3-extern1 | ospfv3-extern2 | ospfv3-inter | ospfv3-intra | rip | ripng | static {address-family [{ipv4-unicast |ipv4-multicast | ipv6-unicast | ipv6-multicast]}
  ```

  Note
  When exporting BGP routes, static routes, configured with the `configure bgp add network` command, take precedence over BGP discovered routes.

For Layer 3 VPNs, a local PE advertises the local customer routes to the remote PE by exporting the BGP routes in the VPN-VRF associated with that customer as VPNv4 routes.

- To enable or disable the export to the remote PE of BGP routes from the CE or static/direct routes in the CE VPN-VRF, use the following commands:

  ```
  enable bgp export [bgp | direct | static] {export-policy} policy-name} {address-family [ipv4-unicast | vpnv4]}
  ```

  ```
  disable bgp export [bgp | direct | static] {address-family [ipv4-unicast | vpnv4]}
  ```

Configure Route Aggregation

Route aggregation is the process of combining the characteristics of several routes so that they are advertised as a single route. Aggregation reduces the amount of information that a *BGP* speaker must store and exchange with other BGP speakers. Reducing the information that is stored and exchanged also reduces the size of the routing table.

- To enable or disable BGP route aggregation, use the following commands:

  ```
  enable bgp aggregation
  ```

  ```
  disable bgp aggregation
  ```

- To create or remove aggregate routes, use the following commands:

  ```
  configure bgp add aggregate-address {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]} ipaddress/masklength {as-match | as-set} {summary-only} {advertise-policy policy} {attribute-policy policy}
  ```

  ```
  configure bgp delete aggregate-address {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast]} [ip address/masklength | all]
  ```

Configure Route Reflectors

The configuration for a router reflector topology takes place on the router or routers that serve as route reflectors. Route reflector clients must be configured for *BGP* participation, but no special configuration is required on route reflector clients to support the route reflector.

- To configure a route reflector to treat neighbors or peer group neighbors as route reflector clients, use the following commands:

  ```
  configure bgp neighbor [remoteaddr | all] [route-reflector-client | no-route-reflector-client]
  ```

  ```
  configure bgp peer-group peer-group-name [route-reflector-client | no-route-reflector-client]
  ```

- If multiple route reflectors are used in a cluster, you must configure the route reflector clients on each route reflector, and you must configure each route reflector with a common cluster ID using the following command:

  ```
  configure bgp cluster-id cluster-id
  ```

Configure a Route Confederation

- To configure every router in the confederation with the same confederation ID, use the following command:

  ```
  configure bgp confederation-id confederation-id
  ```

- For each EBGP confederation peer, use the following command to configure the remote AS number as a confederation sub-AS-number:

  ```
  configure bgp add confederation-peer sub-AS-number number
  ```

- To remove the sub-AS-number configuration for an EBGP peer, use the following command:

  ```
  configure bgp delete confederation-peer sub-AS-number number
  ```

Managing BGP

Enable and Disable BGP Neighbors

- To enable or disable a *BGP* neighbor, use the following commands:

  ```
  enable bgp neighbor [remoteaddr | all]
  disable bgp neighbor [remoteaddr | all]
  ```

Enable and Disable a Peer Group

- To enable or disable a peer group, use the following commands:

  ```
  enable bgp peer-group peer-group-name
  disable bgp peer-group peer-group-name
  ```

Enable and Disable BGP

- To enable or disable *BGP* on the switch, use the following commands:

  ```
  enable bgp
  disable bgp
  ```

Refresh BGP Routes

- To refresh the routes for a *BGP* neighbor, use the following commands:

  ```
  configure bgp neighbor [remoteaddr | all] {address-family [ipv4-
  unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]}
  soft-reset {in | out}
  ```

  ```
  configure bgp peer-group peer-group-name {address-family [ipv4-unicast
  | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4]} soft-reset
  {in | out}
  ```

Configure BFD for BGP

- BFD protection of *BGP* peering sessions allows for the rapid detection of link failures such that peering sessions can be taken out of the established state within fractions of a second.

  ```
  configure bgp {neighbor [all | remoteaddr ]} {bfd [on | off]}
  ```

  Note
  The BFD setting can be applied on a per peer basis, the ability to set BFD on a peer-group or address-family basis is not currently supported. The BGP peer must be in the disabled admin state to modify its BFD setting. While BFD can be enabled on any BGP peering session, protection is only be provided for directly connected EBGP/IBGP peering sessions.

  Note
  Configuring both BFD and graceful restart for BGP on the same switch is counterproductive. When an interface goes down, BFD detects this instantly, stops traffic forwarding and the BGP session goes down, whereas graceful restart forwards traffic despite the interface failure. This behavior might cause traffic loss. Therefore we do not recommend configuring both BFD and graceful restart on the same switch.

Reapply a Policy

- To reapply the route policy associated with the network command, aggregation, import, and redistribution, use the following command:

  ```
  configure bgp soft-reconfiguration
  ```

Clear BGP Flap, Session, or Route Statistics

- To clear the *BGP* flap statistics, use the following command:

  ```
  clear bgp {neighbor} remoteaddr {address-family [ipv4-unicast | ipv4-
  multicast | ipv6-unicast | ipv6-multicast | vpnv4]} flap-statistics
  [all | rd rd_value | as-path path expression | community [no-advertise
  | no-export | no-export-subconfed | number community_num | AS_Num:Num]
  | network [any / netMaskLen | networkPrefixFilter] {exact}]
  ```

  When clearing CE to PE peer sessions, select an IPv4 address family. When clearing PE to PE sessions, select the VPNv4 address family.

Clear BGP Neighbor Counters

- To clear the counters for a *BGP* neighbor, use the following command:

  ```
  clear bgp neighbor [remoteaddr | all] counters
  ```

Displaying BGP Information

Display BGP Router Configuration and Route Statistics

- To display *BGP* router configuration and route statistics, use the following command:

  ```
  show bgp
  ```

Display Peer Group Configuration Information

- To display peer group configuration information, use the following command:

  ```
  show bgp peer-group {detail | peer-group-name {detail}}
  ```

Display BGP Route Information

- To display summary route information, enter the following command:

  ```
  show bgp routes {address-family [ipv4-unicast | ipv4-multicast |ipv6-unicast | ipv6-multicast | vpnv4 | ipv4-vxlan]} summary {vr vr_name}
  ```

- To display route information and statistics, enter the following command:

  ```
  show bgp routes {address-family [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | ipv4-vxlan | {l2vpn-evpn [inclusive-multicast | mac-ip | auto-discovery | esi | ip-prefix]}]} {detail} [ipv4-vxlan | all | as-path path-expression | community [no-advertise | no-export | no-export-subconfed | number community_number | autonomous-system-idbgp-community] | network [any/netMaskLen | networkPrefixFilter] {exact}] {vr vr_name}
  ```

  ```
  show bgp routes address-family vpnv4 {detail} [all | as-path path-expression | community [no-advertise | no-export | no-export-subconfed | number community-number |autonomous-system-idbgp-community] |rd rd network [any/netMaskLen |networkPrefixFilter] {exact}] {vr vr_name}
  ```

- To display information about routes exchanged with a neighbor, enter the following command:

  ```
  show bgp {neighbor} remoteaddr {address-family [ipv4-unicast | ipv4-multicast |ipv6-unicast | ipv6-multicast | ipv4-vxlan | {l2vpn-evpn [inclusive-multicast | mac-ip | auto-discovery | esi | ip-prefix]}]} [accepted-routes | received-routes | rejected-routes | transmitted-routes] {detail} [all | as-path path-expression | community [no-advertise | no-export | no-export-subconfed | number community_number | autonomous-system-id : bgp-community] | network [any/netMaskLen | networkPrefixFilter] {exact}]
  ```

  ```
  show bgp {neighbor} remoteaddr address-family vpnv4 [accepted-routes | received-routes | rejected-routes | transmitted-routes] {detail} [all | as-path path-expression | community [no-advertise | no-export | no-export-subconfed | number community_number | autonomous-
  ```

system-id :*bgp-community*] | **rd** *rd_value* **network** [**any/***netMaskLen* | *networkPrefixFilter*] {**exact**}]

- To display information about suppressed or dampened routes, enter the following command:

  show bgp {**neighbor**} *remoteaddr* {**address-family** [**ipv4-unicast** | **ipv4-multicast** |**ipv6-unicast** | **ipv6-multicast** | **ipv4-vxlan** | {**l2vpn-evpn** [**inclusive-multicast** | **mac-ip** | **auto-discovery** | **esi** | **ip-prefix**]}]} [**accepted-routes** | **received-routes** | **rejected-routes** | **transmitted-routes**] {**detail**} [**all** | **as-path** *path-expression* | **community** [**no-advertise** | **no-export** | **no-export-subconfed** | **number** *community_number* | *autonomous-system-id* : *bgp-community*] | **network** [**any/***netMaskLen* | **networkPrefixFilter**] {**exact**}]

  show bgp {**neighbor**} *remoteaddr* **address-family vpnv4** [**flap-statistics** | **suppressed-routes**] {**detail**} [**all** | **as-path** *path-expression* | **community** [**no-advertise** | **no-export** | **no-export-subconfed** | **number** *community-number* | *autonomous-system-id* : *bgp-community*] | **rd** *rd_value* **network** [**any/** *netMaskLen* | *networkPrefixFilter*] {**exact**}]

Display Layer 3 VPN Peer Session Information

Layer 3 VPN *BGP* session information can be configured with the same commands used for other BGP sessions. When displaying information for CE to PE peer sessions, select an IPv4 address family. When displaying information for PE to PE sessions, select the VPNv4 address family.

- Use the following commands to display Layer 3 VPN session information:

  show bgp routes vpnv4 [**all** | **as-path** | **community** | **detail** | **rd** | **summary**]

Display BGP Memory Usage

- To display *BGP* memory usage information, enter the following command:

  show bgp memory {**detail** | *memoryType*}

Configuration Examples

BGP IPv6 Example

Figure 30 shows the network topology for this example.

Figure 30: BGP IPv6 Example

Switch1 Configuration

```
create vlan v1
configure v1 ipaddress 2001:db8:1::1/48
configure v1 add ports 1
create vlan net
configure net add ports 3
configure net ipaddress 2001:db8:22::1/48
enable ipforwarding ipv6
configure bgp AS-number 100
configure bgp routerid 1.1.1.1
create bgp neighbor 2001:db8:2131::2 remote-AS-number 100
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
configure bgp add network address-family ipv6-unicast 2001:db8:22::/48
configure iproute add 2001:db8:2131::/48 2001:db8:1::2
configure ospfv3 routerid 1.1.1.1
configure ospfv3 add vlan v1 area 0.0.0.0
enable ospfv3
```

Switch2 Configuration

```
create vlan v1
create vlan v2
create vlan lpback
enable loopback lpback
configure v1 ipaddress 2001:db8:1::2/48
configure v2 ipaddress 2001:db8:2::1/48
configure lpback  ipaddress 2001:db8:2131::2/48
enable ipforwarding ipv6
configure v1 add ports 1:9
configure v2 add ports 1:11
configure bgp AS-number 100
configure bgp routerid 1.1.1.2
create bgp neighbor 2001:db8:1::1 remote-AS-number 100
create bgp neighbor 2001:db8:2::2 remote-AS-number 200
configure bgp neighbor 2001:db8:1::1 source-interface ipaddress 2001:db8:2131::2
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
configure ospfv3 routerid 1.1.1.2
configure ospfv3 add vlan all area 0.0.0.0
enable ospfv3
```

Switch3 Configuration

```
create vlan v1
create vlan net
configure v1 add ports 2:11
configure net add ports 1:23
configure v1 ipaddress 2001:db8:2::2/48
configure net ipaddress 2001:db8:55::1/48
enable ipforwarding ipv6
configure bgp AS-number 200
configure bgp routerid 2.1.1.2
create bgp neighbor 2001:db8:2::1 remote-AS-number 100
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
enable bgp export static address-family ipv6-unicast
enable bgp export direct  address-family ipv6-unicast
configure iproute add 2001:db8:66::/48 2001:db8:55::100
```

Configuration Displays (show Commands) for Switch2

```
* Switch2.47 # show bgp neighbor
Peer                AS    Weight State        InMsgs OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------
Ie-- 2001:db8:1::1  100   1      ESTABLISHED  17     19    (0    ) 0:0:11:08
Ee-- 2001:db8:2::2  200   1      ESTABLISHED  28     24    (0    ) 0:0:01:53
Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
(m) EBGP multihop, (r) route reflector client
BGP Peer Statistics
Total Peers       : 2
EBGP Peers        : 1                IBGP Peers          : 1
RR Client         : 0                EBGP Multihop       : 0
Enabled           : 2                Disabled            : 0


* Switch2.50 # show bgp routes address-family ipv6-unicast all
Routes:
Destination             LPref Weight MED  Peer           Next-Hop       AS-Path
------------------------------------------------------------------------------
* ?  2001:db8:2::/48    100   1      0    2001:db8:2::2  2001:db8:2::2  200
*>?  2001:db8:55::/48   100   1      0    2001:db8:2::2  2001:db8:2::2  200
*>?  2001:db8:66::/48   100   1      0    2001:db8:2::2  2001:db8:2::2  200
*>i  2001:db8:22::/48   100   1      0    2001:db8:1::1  2001:db8:1::1
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 4
Feasible Routes   : 4
Active Routes     : 3
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 1
Routes from Ext Peer: 3


Switch2.57 # show bgp neighbor 2001:db8:1::1 address-family ipv6-unicast received-routes
all
Routes:
Destination            LPref Weight MED  Peer           Next-Hop      AS-Path
------------------------------------------------------------------------------
*>i  2001:db8:22::/48  100   1      0    2001:db8:1::1  2001:db8:1::1
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 1
Feasible Routes   : 1
Active Routes     : 1
Rejected Routes   : 0
Unfeasible Routes : 0

Switch2.59 # show bgp routes address-family ipv6-unicast detail all
Routes:
Route: 2001:db8:2::/48, Peer 2001:db8:2::2, BEST
Origin Incomplete, Next-Hop 2001:db8:2::2, LPref 100, MED 0
Weight 1,
As-PATH: 200

Route: 2001:db8:55::/48, Peer 2001:db8:2::2, BEST, Active
Origin Incomplete, Next-Hop 2001:db8:2::2, LPref 100, MED 0
Weight 1,
```

```
As-PATH: 200

Route: 2001:db8:66::/48, Peer 2001:db8:2::2, BEST, Active
Origin Incomplete, Next-Hop 2001:db8:2::2, LPref 100, MED 0
Weight 1,
As-PATH: 200

Route: 2001:db8:22::/48, Peer 2001:db8:1::1, BEST, Active
Origin IGP, Next-Hop 2001:db8:1::1, LPref 100, MED 0
Weight 1,
As-PATH:
BGP Route Statistics
Total Rxed Routes : 4
Feasible Routes   : 4
Active Routes     : 3
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 1
Routes from Ext Peer: 3

Switch2.72 # sh bgp neighbor 2001:db8:2::2 address-family ipv6-unicast transmitted-routes
all
Advertised Routes:
Destination           LPref Weight MED  Peer       Next-Hop         AS-Path
------------------------------------------------------------------------------
>i  2001:db8:22::/48        0      0               2001:db8:2::1    100
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Advertised Routes : 1
```

Graceful BGP Restart Configuration Example for IPv4

In the following IPv4 configuration example, EXOS-1 is the restarting *BGP* router, and EXOS-2 is the receiving BGP router.

To configure router EXOS-1, use the following commands:

```
create vlan bgp-restart
configure vlan bgp-restart add port 2:2
configure vlan bgp-restart ipaddress 10.0.0.1/24
enable ipforwarding

configure bgp as-number 100
configure bgp route-id 10.0.0.1
configure bgp restart both
create bgp neighbor 10.0.0.2 remote-as 64500
enable bgp neighbor all
enable bgp
```

To configure router EXOS-2, use the following commands:

```
create vlan bgp-restart
configure vlan bgp-restart add port 2:5
configure vlan bgp-restart ipaddress 10.0.0.2/24
enable ipforwarding

configure bgp as-number 64500
configure bgp route-id 10.0.0.2
configure bgp restart aware-only
```

```
create bgp neighbor 10.0.0.1 remote-as 100
enable bgp neighbor all
enable bgp
```

Graceful BGP Restart Configuration Example for IPv6

In the following IPv6 configuration example, EXOS-1 is the restarting *BGP* router, and EXOS-2 is the receiving BGP router.

To configure router EXOS-1, use the following commands:

```
create vlan bgp-restart
configure vlan bgp-restart add port 2:2
configure vlan bgp-restart ipaddress 2001:db8:1::1/48
enable ipforwarding ipv6

configure bgp as-number 100
configure bgp routerid 10.0.0.1
configure bgp restart both
conf bgp restart add address-family ipv6-unicast
create bgp neighbor 2001:db8:1::2 remote-as 200
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

To configure router EXOS-2, use the following commands:

```
create vlan bgp-restart
configure vlan bgp-restart add port 2:5
configure vlan bgp-restart ipaddress 2001:db8:1::2/48
enable ipforwarding ipv6
configure bgp as-number 200
configure bgp routerid 10.0.0.2
configure bgp restart aware-only
create bgp neighbor 2001:db8:1::1 remote-as 100
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

Route Reflector Example for IPv4

Figure 27 on page 428 shows the network topology for this example. Router 1.1.1.1 in this example is a regular *BGP* peer.

To configure router 1.1.1.1 to connect to the route reflector as a regular BGP peer, use the following commands:

```
create vlan to_rr
configure vlan to_rr add port 1:1
configure vlan to_rr ipaddress 10.0.0.1/24
enable ipforwarding vlan to_rr

configure bgp router 1.1.1.1
configure bgp as-number 100
create bgp neighbor 10.0.0.2 remote-as 100
enable bgp
enable bgp neighbor all
```

To configure router 2.2.2.2, the route reflector, use the following commands:

```
create vlan to_nc
configure vlan to_nc add port 1:1
configure vlan to_nc ipaddress 10.0.0.1/24
enable ipforwarding vlan to_nc
create vlan to_c1
configure vlan to_c1 add port 1:2
configure vlan to_c1 ipaddress 10.0.2.2/24
enable ipforwarding vlan to_c1

create vlan to_c2
configure vlan to_c2 add port 1:2
configure vlan to_c2 ipaddress 10.0.3.2/24
enable ipforwarding vlan to_c2

configure bgp router 2.2.2.2
configure bgp as-number 100
create bgp neighbor 10.0.0.1 remote-as 100
create bgp neighbor 10.0.2.1 remote-as 100
create bgp neighbor 10.0.3.1 remote-as 100
configure bgp neighbor 10.0.2.1 route-reflector-client
configure bgp neighbor 10.0.3.1 route-reflector-client
enable bgp neighbor all
enable bgp
```

To configure router reflector client 3.3.3.3, use the following commands:

```
create vlan to_rr
configure vlan to_rr add port 1:1
configure vlan to_rr ipaddress 10.0.2.1/24
enable ipforwarding vlan to_rr

configure bgp router 3.3.3.3
configure bgp as-number 100
create bgp neighbor 10.0.2.2 remote-as 100
enable bgp neighbor all
enable bgp
```

To configure route reflector client 4.4.4.4, use the following commands:

```
create vlan to_rr
configure vlan to_rr add port 1:1
configure vlan to_rr ipaddress 10.0.3.1/24
enable ipforwarding vlan to_rr

configure bgp router 4.4.4.4
configure bgp as-number 100
create bgp neighbor 10.0.3.2 remote-as 100
enable bgp neighbor all
enable bgp
```

Route Reflector Example for IPv6

Figure 31 shows the network topology for this example.

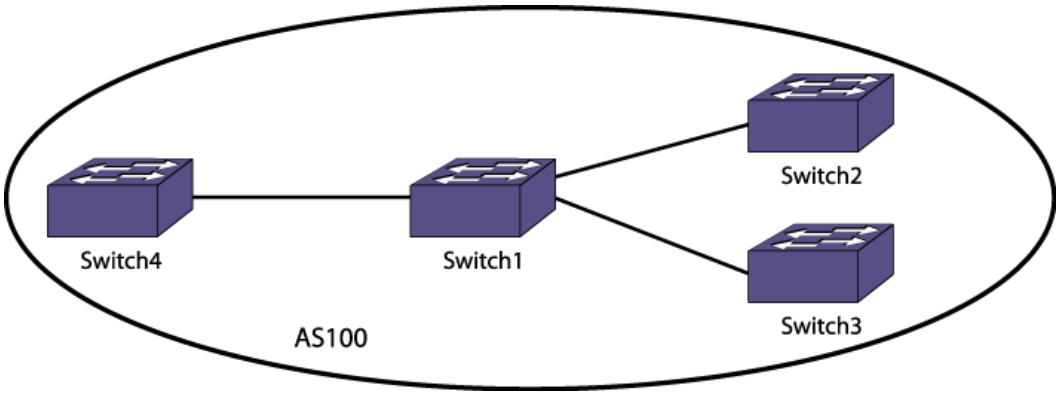


Figure 31: Route Reflector Example

Switch1 is the route reflector, and Switch2 and Switch3 are route reflector clients. Switch4 is a regular neighbor (non-route-reflector client).

Route Reflector (Switch1) Configuration

To configure the route reflector, enter the following commands

```
create vlan "v3"
create vlan "v1"
create vlan "v2"
configure vlan v3 add ports 1:16
configure vlan v1 add ports 1:9
configure vlan v2 add ports 1:1
configure v1 ipaddress 2001:db8:1::1/48
configure v2 ipaddress 2001:db8:3::1/48
configure v3 ipaddress 2001:db8:5::1/484
enable ipforwarding ipv6
configure bgp AS-number 100
configure bgp routerid 1.1.1.1
create bgp neighbor 2001:db8:1::2 remote-AS-number 100
configure bgp neighbor 2001:db8:1::2 route-reflector-client
create bgp neighbor 2001:db8:3::2 remote-AS-number 100
configure bgp neighbor 2001:db8:3::2 route-reflector-client
create bgp neighbor 2001:db8:5::2 remote-AS-number 100
enable bgp neighbor 2001:db8:1::2 capability ipv6-unicast
enable bgp neighbor 2001:db8:3::2 capability ipv6-unicast
enable bgp neighbor 2001:db8:5::2 capability ipv6-unicast
enable bgp neighbor all
enable bgp
configure ospfv3 routerid 1.1.1.1
configure ospfv3 add vlan all area 0.0.0.0
enable ospfv3
```

Switch2 Route Reflector Client Configuration

To configure the Switch 2 route reflector client, enter the following commands:

```
create vlan "v1"
configure vlan v1 add ports 1
configure v1 ipaddress 2001:db8:1::2/48
enable ipforwarding ipv6 vlan v1
configure bgp AS-number 100
configure bgp routerid 1.1.1.2
create bgp neighbor 2001:db8:1::1 remote-AS-number 100
enable bgp neighbor 2001:db8:1::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:1::1
enable bgp
```

```
configure ospfv3 routerid 1.1.1.2
configure ospfv3 add vlan v1 area 0.0.0.0
enable ospfv3
```

Switch3 Route Reflector Client Configuration

To configure the Switch 3 route reflector client,e nter the following commands:

```
create vlan "v1"
configure vlan v1 add ports 1
configure v1 ipaddress 2001:db8:3::2/48
enable ipforwarding ipv6
configure bgp AS-number 100
configure bgp routerid 2.1.1.2
create bgp neighbor 2001:db8:3::1 remote-AS-number 100
enable bgp neighbor 2001:db8:3::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:3::1
enable bgp
configure ospfv3 routerid 2.1.1.2
configure ospfv3 add vlan v1 area 0.0.0.0
enable ospfv3
```

Switch4 Configuration

Switch4 is not a route reflector client. To configure this switch, enter the following commands:

```
create vlan "net"
enable loopback-mode vlan net
create vlan "v1"
configure vlan v1 add ports 9
configure v1 ipaddress 2001:db8:5::2/48
configure net ipaddress 2001:db8:2555::1/48
enable ipforwarding ipv6
configure bgp AS-number 100
configure bgp routerid 5.1.1.2
configure bgp add network address-family ipv6-unicast 2001:db8:2555::/48
create bgp neighbor 2001:db8:5::1 remote-AS-number 100
enable bgp neighbor 2001:db8:5::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:5::1
enable bgp
configure ospfv3 routerid 5.1.1.2
configure ospfv3 add vlan v1 area 0.0.0.0
enable ospfv3
```

Configuration Display for Switch1

The following display shows that the route reflector has two route reflector client peers (r flag), and one regular peer:

```
* Switch1.30 # show bgp neighbor
Peer            AS    Weight State        InMsgs OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------
Ier- 2001:db8:1::2   100   1      ESTABLISHED  6      8     (0    ) 0:0:04:03
Ier- 2001:db8:3::2   100   1      ESTABLISHED  5      7     (0    ) 0:0:02:51
Ie-- 2001:db8:5::2   100   1      ESTABLISHED  6      7     (0    ) 0:0:01:41
Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
(m) EBGP multihop, (r) route reflector client
BGP Peer Statistics
Total Peers       : 3
EBGP Peers        : 0                IBGP Peers          : 3
RR Client         : 2                EBGP Multihop       : 0
Enabled           : 3                Disabled            : 0
```

Configuration Display for Switch2

The following command displays the *BGP* routes for route reflector client Switch2:

```
* Switch2.10 # show bgp routes address-family ipv6-unicast detail all
Routes:
Route: 2001:db8:2555::/48, Peer 2001:db8:1::1, BEST, Active
Origin IGP, Next-Hop 2001:db8:5::2, LPref 100, MED 0
Weight 1, RR Orig ID 5.1.1.2
As-PATH:
RR Cluster IDs: 1.1.1.1
BGP Route Statistics
Total Rxed Routes : 1
Feasible Routes   : 1
Active Routes     : 1
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 1
Routes from Ext Peer: 0
```

Route Confederation Example for IPv4

The following figure shows the network topology for this example.

To configure router A, use the following commands:

```
create vlan ab
configure vlan ab add port 1
configure vlan ab ipaddress 10.1.1.6/30
enable ipforwarding vlan ab
configure ospf add vlan ab area 0.0.0.0

create vlan ac
configure vlan ac add port 2
configure vlan ac ipaddress 10.1.1.17/30
enable ipforwarding vlan ac
configure ospf add vlan ac area 0.0.0.0
enable ospf

configure bgp as-number 64505
configure bgp routerid 10.1.1.17
configure bgp confederation-id 64500
enable bgp

create bgp neighbor 10.1.1.5 remote-AS-number 64505
create bgp neighbor 10.1.1.18 remote-AS-number 64505
enable bgp neighbor all
```

To configure router B, use the following commands:

```
create vlan ba
configure vlan ba add port 1
configure vlan ba ipaddress 10.1.1.5/30
enable ipforwarding vlan ba
configure ospf add vlan ba area 0.0.0.0

create vlan bc
configure vlan bc add port 2
configure vlan bc ipaddress 10.1.1.22/30
enable ipforwarding vlan bc
configure ospf add vlan bc area 0.0.0.0
```

```
create vlan bd
configure vlan bd add port 3
configure vlan bd ipaddress 10.1.1.9/30
enable ipforwarding vlan bd
configure ospf add vlan bd area 0.0.0.0
enable ospf

configure bgp as-number 64505
configure bgp routerid 10.1.1.22
configure bgp confederation-id 64500
enable bgp

create bgp neighbor 10.1.1.6 remote-AS-number 64505
create bgp neighbor 10.1.1.21 remote-AS-number 64505
create bgp neighbor 10.1.1.10 remote-AS-number 64510
configure bgp add confederation-peer sub-AS-number 64510
enable bgp neighbor all
```

To configure router C, use the following commands:

```
create vlan ca
configure vlan ca add port 1
configure vlan ca ipaddress 10.1.1.18/30
enable ipforwarding vlan ca
configure ospf add vlan ca area 0.0.0.0

create vlan cb
configure vlan cb add port 2
configure vlan cb ipaddress 10.1.1.21/30
enable ipforwarding vlan cb
configure ospf add vlan cb area 0.0.0.0
enable ospf

configure bgp as-number 64505
configure bgp routerid 10.1.1.21
configure bgp confederation-id 64500
enable bgp

create bgp neighbor 10.1.1.22 remote-AS-number 64505
create bgp neighbor 10.1.1.17 remote-AS-number 64505
enable bgp neighbor all
```

To configure router D, use the following commands:

```
create vlan db
configure vlan db add port 1
configure vlan db ipaddress 10.1.1.10/30
enable ipforwarding vlan db
configure ospf add vlan db area 0.0.0.0

create vlan de
configure vlan de add port 2
configure vlan de ipaddress 10.1.1.14/30
enable ipforwarding vlan de
configure ospf add vlan de area 0.0.0.0
enable ospf

configure bgp as-number 64510
configure bgp routerid 10.1.1.14
configure bgp confederation-id 64500
enable bgp

create bgp neighbor 10.1.1.9 remote-AS-number 64505
```

```
create bgp neighbor 10.1.1.13 remote-AS-number 64510
configure bgp add confederation-peer sub-AS-number 64505
enable bgp neighbor all
```

To configure router E, use the following commands:

```
create vlan ed
configure vlan ed add port 1
configure vlan ed ipaddress 10.1.1.13/30
enable ipforwarding vlan ed
configure ospf add vlan ed area 0.0.0.0
enable ospf

configure bgp as-number 64510
configure bgp routerid 10.1.1.13
configure bgp confederation-id 64500
enable bgp

create bgp neighbor 10.1.1.14 remote-AS-number 64510
enable bgp neighbor 10.1.1.14
```

Route Confederation Example for IPv6

Figure 28 on page 429 shows the network topology for this example.

To configure router A, use the following commands:

```
create vlan ab
configure vlan ab add port 23
configure vlan ab ipaddress 2001:db8:1::6/48
enable ipforwarding ipv6 vlan ab
configure ospf add vlan ab area 0.0.0.0
create vlan ac
configure vlan ac add port 16
configure vlan ac ipaddress 2001:db8:3::17/48
enable ipforwarding ipv6 vlan ac
configure ospfv3 routerid 10.1.1.17
configure ospfv3 add vlan ac area 0.0.0.0
enable ospfv3
configure bgp as-number 65001
configure bgp routerid 10.1.1.17
configure bgp confederation-id 200
create bgp neighbor 2001:db8:1::5 remote-AS-number 65001
create bgp neighbor 2001:db8:3::18 remote-AS-number 65001
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

To configure router B, use the following commands:

```
configure default delete port all
create vlan ba
configure vlan ba add port 5:33
configure vlan ba ipaddress 2001:db8:1::5/48
enable ipforwarding ipv6 vlan ba
configure ospf add vlan ba area 0.0.0.0
create vlan bc
configure vlan bc add port 5:8
configure vlan bc ipaddress 2001:db8:2::22/48
enable ipforwarding ipv6 vlan bc
configure ospf add vlan bc area 0.0.0.0
create vlan bd
configure vlan bd add port 5:4
```

```
configure vlan bd ipaddress 2001:db8:4::9/48
enable ipforwarding ipv6 vlan bd
configure ospfv3 add vlan bd area 0.0.0.0
configure ospfv3 routerid 10.1.1.22
enable ospfv3
configure bgp as-number 65001
configure bgp routerid 10.1.1.22
configure bgp confederation-id 200
create bgp neighbor 2001:db8:1::6 remote-AS-number 65001
create bgp neighbor 2001:db8:2::21 remote-AS-number 65001
create bgp neighbor 2001:db8:4::10 remote-AS-number 65002
configure bgp add confederation-peer sub-AS-number 65002
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

To configure router C, use the following commands:

```
configure default delete port all
create vlan ca
configure vlan ca add port 2:16
configure vlan ca ipaddress 2001:db8:3::18/48
enable ipforwarding ipv6 vlan ca
configure ospfv3 add vlan ca area 0.0.0.0
create vlan cb
configure vlan cb add port 2:5
configure vlan cb ipaddress 2001:db8:2::21/48
enable ipforwarding ipv6 vlan cb
configure ospfv3 add vlan cb area 0.0.0.0
configure ospfv3 routerid 10.1.1.21
enable ospfv3
configure bgp as-number 65001
configure bgp routerid 10.1.1.21
configure bgp confederation-id 200
create bgp neighbor 2001:db8:2::22 remote-AS-number 65001
create bgp neighbor 2001:db8:3::17 remote-AS-number 65001
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

To configure router D, use the following commands:

```
configure default delete port all
create vlan db
configure vlan db add port 13
configure vlan db ipaddress 2001:db8:4::10/48
enable ipforwarding ipv6 vlan db
configure ospf add vlan db area 0.0.0.0
create vlan de
configure vlan de add port 5
configure vlan de ipaddress 2001:db8:5::14/48
enable ipforwarding ipv6 vlan de
configure ospfv3 add vlan de area 0.0.0.0
configure ospfv3 routerid 10.1.1.14
enable ospfv3
configure bgp as-number 65002
configure bgp routerid 10.1.1.14
configure bgp confederation-id 200
create bgp neighbor 2001:db8:4::9 remote-AS-number 65001
create bgp neighbor 2001:db8:5::13 remote-AS-number 65002
configure bgp add confederation-peer sub-AS-number 65001
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

To configure router E, use the following commands:

```
configure default delete port all
create vlan ed
configure vlan ed add port 5
configure vlan ed ipaddress 2001:db8:5::13/48
enable ipforwarding ipv6 vlan ed
configure ospfv3 add vlan ed area 0.0.0.0
configure ospfv3 routerid 10.1.1.13
enable ospfv3
configure bgp as-number 65002
configure bgp routerid 10.1.1.13
configure bgp confederation-id 200
create bgp neighbor 2001:db8:5::14 remote-AS-number 65002
enable bgp neighbor all capability ipv6-unicast
enable bgp neighbor all
enable bgp
```

Default Route Origination Example for IPv4

The following example configures the originate default route feature for *BGP* neighbor 10.203.134.5 using policy def_originate.pol.

```
def_originate.pol
entry prefix_matching {
if match any {
nlri 192.168.3.0/24;
} then {
as-path "64505";
permit;
}
}
enable bgp neighbor 10.203.134.5 originate-default policy def_originate
```

With this configuration, a default route is originated and sent to neighbor 10.203.134.5 only if there is a BGP route in the local RIB which matches the statement nlri 192.168.3.0/24. If a matching route exists, the default route is sent to neighbor 10.203.134.5 with the 64505 as-path prepended. If this is an EBGP neighbor, then the local AS-Number is prepended after 64505.

If the route for the match statement nlri 192.168.3.0/24 goes away and there is no other matching route in the BGP RIB, the default route origination feature becomes inactive and BGP withdraws the 0.0.0.0/0 default route from neighbor 10.203.134.5. When a matching route becomes available again in the local BGP RIB, the default route origination feature becomes active again and the default route 0.0.0.0/0 is advertised to neighbor 10.203.134.5.

Default Route Origination Example for IPv6

The following example configures the originate default route feature for *BGP* neighbor 2001:db8:1::2 using policy def_originate.pol.

```
def_originate.pol
entry prefix_matching {
if match any {
nlri 2001:db8:2::/48;
} then {
```

```
as-path "65001";
permit;
}
}
enable bgp neighbor 2001:db8:1::2 address-family ipv6-unicast originate-default policy
def_originate
```

With this configuration, a default route is originated and sent to neighbor 2001:db8:1::2 only if there is a BGP route in the local RIB which matches the statement nlri 2001:db8:2::/48. If a matching route exists, the default route is sent to neighbor 2001:db8:1::2 with the 65001 as-path prepended. If this is an EBGP neighbor, then the local AS-Number is prepended after 65001.

If the route for the match statement 2001:db8:2::/48 goes away and there is no other matching route in the BGP RIB, the default route origination feature becomes inactive and BGP withdraws the default route ::/0 from neighbor 2001:db8:1::2. When a matching route becomes available again in the local BGP RIB, the default route origination feature becomes active again and the default route ::/0 is advertised to neighbor 2001:db8:1::2.

BGP Speaker Black Hole Example

Black hole routing is used to protect a service provider's internal network from distributed denial of service (DDoS) attacks. The strategy is to drop inbound DDoS attack traffic destined to a target network at the edge of the provider network as soon as the target is identified.

Since the attack traffic may enter the service provider network from any of its edge routers, it is not feasible to manually configure a static black hole route entry on each edge router. The problem is further complicated by the fact that the target network, and the need to block traffic to it, is dynamic. Also, the service provider may serve multiple customers and you don't want to drop traffic to a customer network until it is identified as a target for an ongoing attack.

Instead, a service provider can use *BGP* to distribute a black hole route (route entry for the target network with a black hole next-hop) from a single router to all its edge BGP speakers that will then drop traffic destined to the victim's network right at the provider edge. Figure 32 shows an example topology to achieve this.

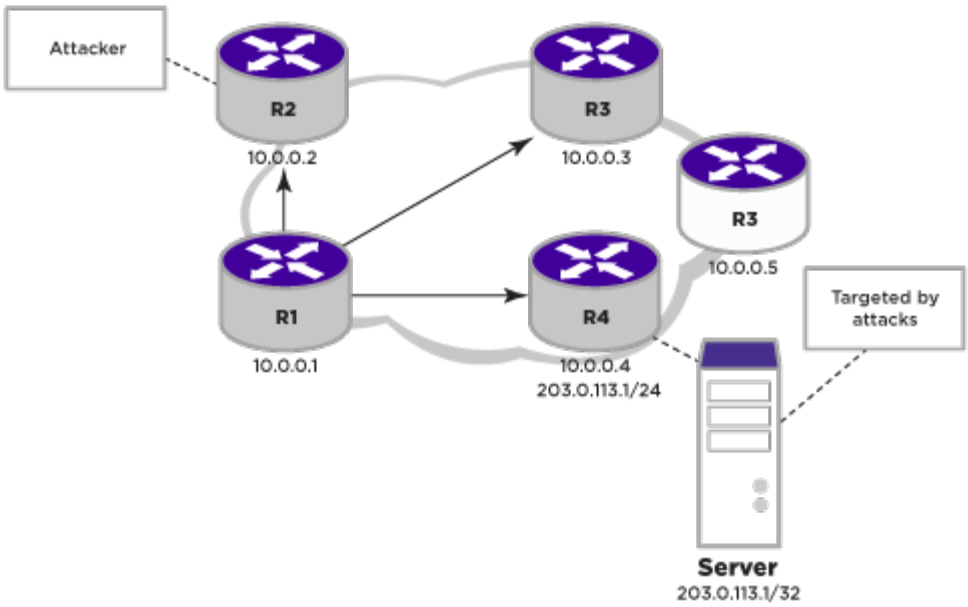


Figure 32: Black Hole Routing Using BGP

Step 1

Prior to the attack, select an address for the intended black hole next-hop. Configure the forwarding plane of each edge router so that packets forwarded to this next-hop are dropped:

1. Create a black hole *VLAN* with an IP address that is in the same subnet as the chosen black hole next-hop.
2. Add an active port to the black hole VLAN (usually an unused port in the switch).
3. Create a static FDB entry that maps a well-chosen, unused MAC address to the black hole VLAN and the active port added to that VLAN.
4. Create a static ARP entry that maps the black hole next-hop to the above MAC address.
5. Create an filter to deny packets that exit the blackhole VLAN.

   In the following example configuration, 192.168.2.0/24 is the subnet of the black hole VLAN, “BH_VLAN,” and 192.168.2.66 is the chosen black hole next-hop. The active port 6:9 is added as the egress port for “BH_VLAN.”

```
create vlan BH_VLAN
configure vlan BH_VLAN tag 666
enable loopback-mode vlan BH_VLAN
configure vlan BH_VLAN ipaddress 192.168.2.1 255.255.255.0
enable ipforwarding vlan BH_VLAN
disable igmp snooping vlan BH_VLAN
disable igmp vlan BH_VLAN
create fdb 00:02:03:04:05:06 vlan BH_VLAN port 6:9
configure iparp add 192.168.2.66 vr VR-Default 00:02:03:04:05:06
configure access-list BH_ACL vlan BH_VLAN egress
```

   When a packet arrives in the forwarding plane and looks up a route that has the above black hole next-hop as its next-hop, a subsequent ARP and *FDB (forwarding database)* look-up occurs that forwards the packet to exit the switch using the above

black hole VLAN, “BH_VLAN,” and port “6:9.” The packet is dropped due to the deny action in the egress ACL filter.

The following policy file discards any traffic that exits the black hole VLAN, “BH_VLAN.” Note that the match on “source-address 0.0.0.0/0” matches any egress packet ensuring that all packets exiting via the black hole VLAN are dropped:

```
edit policy BH_ACL
entry bh-acl {
if {
    source-address 0.0.0.0/0;
} then {
deny ;
}
}
```

Step 2

- Prior to the attack, configure inbound route-maps on all edge *BGP* speakers (R2 through R4 in Figure 32 on page 469).

  These inbound policies modify the next-hop of specifically marked BGP network layer reach-ability information (NLRIs) to point to the chosen black hole next-hop. We use BGP community or extended-community attributes to identify NLRIs that need to be black holed (ones whose next-hops have to be modified). The community values that are chosen should be reserved for this purpose within the provider network.

  In the following example, a community of 666:0 is chosen for identifying blackhole routes. The next-hop of BGP NLRIs with that community attribute is modified to use the blackhole next-hop.

```
R3.1 # edit policy BH_policy_NH
entry bh-nhset {
        if match any {
            community 666:0;
            nlri any/32 ;
        } then {
            next-hop 192.168.2.66 ;
            permit ;
        }
}
entry bh-default {
        if match any {
        } then {
            permit ;
        }
}
```

Step 3

Once the target network has been identified during a DDoS attack, apply an outbound policy or export policy to one router (in our example, R1) within the provider network so that the route to the target network is advertised to the other edge routers within the community 666:0.

The following example creates a static route on R1 to the target network 203.0.113.1/32 with a static export policy that applies to the community. When the attack targets change, you only need to create or delete static routes to the target networks. The

policy exports them to the edge *BGP* speakers with the selected community attribute values attached.

```
R1.1 # edit policy BH_COMM_APPLY
entry bh-comm-apply {
        if match any  {
            nlri 203.0.113.0/24;
            nlri any/32;
        } then {
            community set “666:0”;
        }
}
R1.2 # configure iproute add 203.0.113.1/32 10.0.0.6
R1.3 # enable bgp export static export-policy BH_COMM_APPLY
```

Alternatively, you can apply the policy as an outbound policy as below:

```
R1.10 # configure bgp neighbor 10.0.0.2 route-policy out BH_COMM_APPLYR1.11
# configure bgp neighbor 10.0.0.3 route-policy out BH_COMM_APPLYR1.12
# configure bgp neighbor 10.0.0.4 route-policy out BH_COMM_APPLY
```

Output

```
R4.67 # show bgp route all
Routes:
```

Destination	Peer	Next-Hop	LPref	Weight	MED	AS-Path
*>i 192.51.100.0/28	10.0.0.1	10.0.0.1	100	1	0	64500
*>i 192.51.100.16/28	10.0.0.1	10.0.0.1	100	1	0	64500
*>i 192.51.100.32/28	10.0.0.1	10.0.0.1	100	1	0	64500
*>i 192.51.100.48/28	10.0.0.1	10.0.0.1	100	1	0	64500
*>i 192.51.100.64/28	10.0.0.1	10.0.0.1	100	1	0	64500
*>i 203.0.113.1/32	10.0.0.1	192.168.2.66	100	1	0	64500

```
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 6
Feasible Routes : 6
```

```
Active Routes : 6
Rejected Routes : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 0
Routes from Ext Peer: 6
Switch.68 # rtlookup 203.0.113.1
Ori  Destination      Gateway         Mtr  Flags         VLAN      Duration
#be  203.0.113.1/32   192.168.2.66    1   UG-D---um--f BH_VLAN   0d:1h:5m:5s
```

Note

For the above solution, the edge routers, R1 through R4, may still export the route to the target network to external AS(s), but the traffic is dropped at the edge of the provider network.

An alternative solution for protecting the network is to perform step 1 only on a designated sink router, (R5 in Figure 32 on page 469) and redistributes the black hole next-hop using iBGP to R2 through R4. When traffic arrives at routers R2 through R4, it is forwarded to R5, since R2–R4 have iBGP routes that resolve the black hole next-hop to R5. Router R5 then discards the traffic.

BGP Route Filtering Example for IPv4

You can use policy files with *BGP* attributes to filter IPv4 routes. The example in this section is for the topology shown in Figure 33.

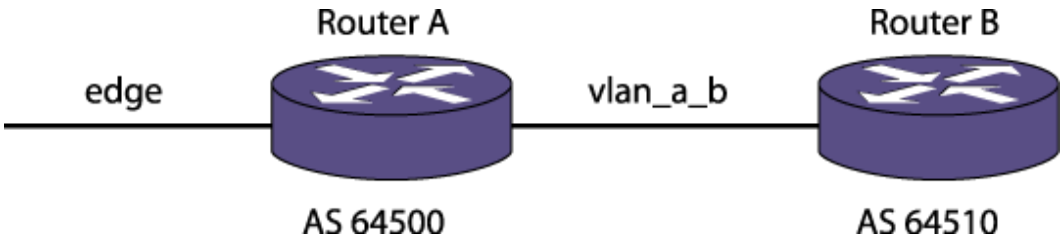


Figure 33: BGP IPv4 Route Filtering Example

Router A Configuration

```
configure vlan default delete ports all
create vlan "a_b"
enable ipforwarding ipv6 vlan a_b
create vlan "edge"
enable ipforwarding ipv6 vlan edge
configure vlan a_b add ports 5 untagged
configure vlan edge add ports 23 untagged
configure edge ipaddress 2001:db8:2000::1/48
configure a_b ipaddress 2001:db8:3000::1/48
configure bgp AS-number 2100
configure bgp routerid 10.10.10.1
enable bgp community format AS-number:number
create bgp neighbor 2001:db8:2000::2 remote-AS-number 1100
enable bgp neighbor 2001:db8:2000::2
create bgp neighbor 2001:db8:3000::2 remote-AS-number 3300
enable bgp neighbor 2001:db8:3000::2
enable bgp neighbor 2001:db8:2000::2 capability ipv6-unicast
enable bgp neighbor 2001:db8:2000::2 address-family ipv6-unicast soft-in-reset
enable bgp neighbor 2001:db8:2000::2 capability ipv6-multicast
```

```
configure bgp neighbor 2001:db8:3000::2 send-community both
enable bgp neighbor 2001:db8:3000::2 capability ipv6-unicast
enable bgp neighbor 2001:db8:3000::2 capability ipv6-multicast
enable bgp
```

Router B Configuration

```
configure vlan default delete ports all
create vlan "b_a"
enable ipforwarding ipv6 vlan b_a
configure vlan b_a add ports 25 untagged
configure b_a ipaddress 2001:db8:3000::2/48
configure bgp AS-number 3300
configure bgp routerid 10.20.20.2
enable bgp community format AS-number:number
create bgp neighbor 2001:db8:3000::1 remote-AS-number 2100
enable bgp neighbor 2001:db8:3000::1
enable bgp neighbor 2001:db8:3000::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:3000::1 capability ipv6-multicast
enable bgp
```

BGP Routes Before Policy Application

The following example shows the routes in the *BGP* routing table at Router A after completing the configuration described in the previous two sections:

```
* Switch.122 # show bgp routes address-family ipv6-unicast all
Routes:
Destination                                  LPref Weight MED
Peer                                    Next-Hop
AS-Path
-------------------------------------------------------------------------------
*>i  2001:db8:4001::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:4002::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:4003::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:5030::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 4977
*>i  2001:db8:5031::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 4977
*>i  2001:db8:a004::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 6
Feasible Routes   : 6
Active Routes     : 6
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
```

```
Routes from Int Peer: 0
Routes from Ext Peer: 6
```

Creating and Applying the Route Filter Policy

The policy described in this section is applied to Router A and does the following:

- Denies routes in network 2001:db8:a004::/48
- Sets the community and MED values for routes in network 2001:db8:4000::/44
- Sets the community and MED values for routes that contain AS path 4977

The following is a route filter policy named custFilter:

```
entry et1 {
if match all {
nlri 2001:db8:a004::/48;
} then {
deny;
}
}
entry et2 {
if match any {
nlri  2001:db8:4000::/44;
as-path 4977;
} then {
med set 100;
community set "2342:6788";
permit;
}
}
```

Apply the custFilter inbound policy.

```
* Switch.53 # configure bgp neighbor 2001:db8:2000::1 ipv6-unicast route-policy in
custfilter
```

BGP Routes After Policy Application

The following example shows the routes in the *BGP* routing table at Router A after applying the custFilter inbound policy:

```
* Switch.127 # show bgp routes address-family ipv6-unicast all
Routes:
Destination                                LPref Weight MED
Peer                                  Next-Hop
AS-Path
------------------------------------------------------------------------------
*>i  2001:db8:4001::/48                          100   1      100
2001:db8:2000::2                      2001:db8:2000::2
1100 5623
*>i  2001:db8:4002::/48                          100   1      100
2001:db8:2000::2                      2001:db8:2000::2
1100 5623
*>i  2001:db8:4003::/48                          100   1      100
2001:db8:2000::2                      2001:db8:2000::2
1100 5623
*>i  2001:db8:5030::/48                          100   1      100
2001:db8:2000::2                      2001:db8:2000::2
1100 4977
```

```
*>i  2001:db8:5031::/48                          100   1      100
2001:db8:2000::2                         2001:db8:2000::2
1100 4977
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 6
Feasible Routes   : 5
Active Routes     : 5
Rejected Routes   : 1
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 0
Routes from Ext Peer: 5
```

Route 2001:db8:a004::/48 is not present in the BGP routing table shown above.

The routes were updated because soft-reset is configured for this neighbor.

The following command examples show that the denied routes are not transmitted to the neighbors:

```
* Switch.130 # show bgp neighbor 2001:db8:3000::2 address-family ipv6-unicast transmitted-
routes all
Advertised Routes:
Destination                                LPref Weight MED
Peer                                     Next-Hop
AS-Path
------------------------------------------------------------------------------
>i  2001:db8:4001::/48                          0            100
2001:db8:3000::1
2100 1100 5623
>i  2001:db8:4002::/48                          0            100
2001:db8:3000::1
2100 1100 5623
>i  2001:db8:4003::/48                          0            100
2001:db8:3000::1
2100 1100 5623
>i  2001:db8:5030::/48                          0            100
2001:db8:3000::1
2100 1100 4977
>i  2001:db8:5031::/48                          0            100
2001:db8:3000::1
2100 1100 4977
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Advertised Routes : 5
```

The next example shows another way to see that the MED values are set as defined in the policy.

```
* Switch.131 # show bgp neighbor 2001:db8:3000::2 address-family ipv6-unicast transmitted-
routes detail all
Advertised Routes:
Route: 2001:db8:4001::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 5623
```

```
Route: 2001:db8:4002::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 5623
Route: 2001:db8:4003::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 5623
Route: 2001:db8:5030::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 4977
Route: 2001:db8:5031::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 4977
BGP Route Statistics
Advertised Routes : 5
```

BGP Route Filtering Example for IPv6

You can use policy files with *BGP* attributes to filter IPv6 routes. The example in this section is for the topology shown in the following figure.

Router A Configuration

```
configure vlan default delete ports all
create vlan "a_b"
enable ipforwarding ipv6 vlan a_b
create vlan "edge"
enable ipforwarding ipv6 vlan edge
configure vlan a_b add ports 5 untagged
configure vlan edge add ports 23 untagged
configure edge ipaddress 2001:db8:2000::1/48
configure a_b ipaddress 2001:db8:3000::1/48
configure bgp AS-number 2100
configure bgp routerid 10.10.10.1
enable bgp community format AS-number:number
create bgp neighbor 2001:db8:2000::2 remote-AS-number 1100
enable bgp neighbor 2001:db8:2000::2
create bgp neighbor 2001:db8:3000::2 remote-AS-number 3300
enable bgp neighbor 2001:db8:3000::2
enable bgp neighbor 2001:db8:2000::2 capability ipv6-unicast
enable bgp neighbor 2001:db8:2000::2 address-family ipv6-unicast soft-in-reset
enable bgp neighbor 2001:db8:2000::2 capability ipv6-multicast
configure bgp neighbor 2001:db8:3000::2 send-community both
enable bgp neighbor 2001:db8:3000::2 capability ipv6-unicast
enable bgp neighbor 2001:db8:3000::2 capability ipv6-multicast
enable bgp
```

Router B Configuration

```
configure vlan default delete ports all
create vlan "b_a"
enable ipforwarding ipv6 vlan b_a
configure vlan b_a add ports 25 untagged
configure b_a ipaddress 2001:db8:3000::2/48
configure bgp AS-number 3300
configure bgp routerid 10.20.20.2
enable bgp community format AS-number:number
create bgp neighbor 2001:db8:3000::1 remote-AS-number 2100
enable bgp neighbor 2001:db8:3000::1
enable bgp neighbor 2001:db8:3000::1 capability ipv6-unicast
```

```
enable bgp neighbor 2001:db8:3000::1 capability ipv6-multicast
enable bgp
```

BGP Routes Before Policy Application

The following example shows the routes in the *BGP* routing table at Router A after completing the configuration described in the previous two sections:

```
* Switch.122 # show bgp routes address-family ipv6-unicast all
Routes:
Destination                                LPref Weight MED
Peer                                    Next-Hop
AS-Path
------------------------------------------------------------------------
*>i  2001:db8:4001::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:4002::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:4003::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:5030::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 4977
*>i  2001:db8:5031::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100 4977
*>i  2001:db8:a004::/48                          100   1      0
2001:db8:2000::2                        2001:db8:2000::2
1100
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 6
Feasible Routes   : 6
Active Routes     : 6
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 0
Routes from Ext Peer: 6
```

Creating and Applying the Route Filter Policy

The policy described in this section is applied to Router A and does the following:

- Denies routes in network 2001:db8:a004::/48
- Sets the community and MED values for routes in network 2001:db8:4000::/44
- Sets the community and MED values for routes that contain AS path 4977

The following is a route filter policy named custFilter:

```
entry et1 {
if match all {
nlri 2001:db8:a004::/48;
} then {
deny;
```

```
}
}
entry et2 {
if match any {
nlri  2001:db8:4000::/44;
as-path 4977;
} then {
med set 100;
community set "2342:6788";
permit;
}
}
```

Apply the custFilter inbound policy.

```
* Switch.53 # configure bgp neighbor 2001:db8:2000::1 ipv6-unicast route-policy in
custfilter
```

BGP Routes After Policy Application

The following example shows the routes in the *BGP* routing table at Router A after applying the custFilter inbound policy:

```
* Switch.127 # show bgp routes address-family ipv6-unicast all
Routes:
Destination                                LPref Weight MED
Peer                                    Next-Hop
AS-Path
------------------------------------------------------------------------------
*>i  2001:db8:4001::/48                          100   1      100
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:4002::/48                          100   1      100
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:4003::/48                          100   1      100
2001:db8:2000::2                        2001:db8:2000::2
1100 5623
*>i  2001:db8:5030::/48                          100   1      100
2001:db8:2000::2                        2001:db8:2000::2
1100 4977
*>i  2001:db8:5031::/48                          100   1      100
2001:db8:2000::2                        2001:db8:2000::2
1100 4977
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 6
Feasible Routes   : 5
Active Routes     : 5
Rejected Routes   : 1
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 0
Routes from Ext Peer: 5
```

Route 2001:db8:a004::/48 is not present in the BGP routing table shown above.

The routes were updated because soft-reset is configured for this neighbor.

The following command examples show that the denied routes are not transmitted to the neighbors:

```
* Switch.130 # show bgp neighbor 2001:db8:3000::2 address-family ipv6-unicast transmitted-
routes all
Advertised Routes:
Destination                                LPref Weight MED
Peer                                    Next-Hop
AS-Path
------------------------------------------------------------------------------
>i  2001:db8:4001::/48                          0            100
2001:db8:3000::1
2100 1100 5623
>i  2001:db8:4002::/48                          0            100
2001:db8:3000::1
2100 1100 5623
>i  2001:db8:4003::/48                          0            100
2001:db8:3000::1
2100 1100 5623
>i  2001:db8:5030::/48                          0            100
2001:db8:3000::1
2100 1100 4977
>i  2001:db8:5031::/48                          0            100
2001:db8:3000::1
2100 1100 4977
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Advertised Routes : 5
```

The next example shows another way to see that the MED values are set as defined in the policy.

```
* Switch.131 # show bgp neighbor 2001:db8:3000::2 address-family ipv6-unicast transmitted-
routes detail all
Advertised Routes:
Route: 2001:db8:4001::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 5623
Route: 2001:db8:4002::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 5623
Route: 2001:db8:4003::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 5623
Route: 2001:db8:5030::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 4977
Route: 2001:db8:5031::/48, Active
Origin IGP, Next-Hop 2001:db8:3000::1, MED 100
As-PATH: 2100 1100 4977
BGP Route Statistics
Advertised Routes : 5
```

Route Aggregation Example for IPV4

Figure 34 shows the topology for this example.

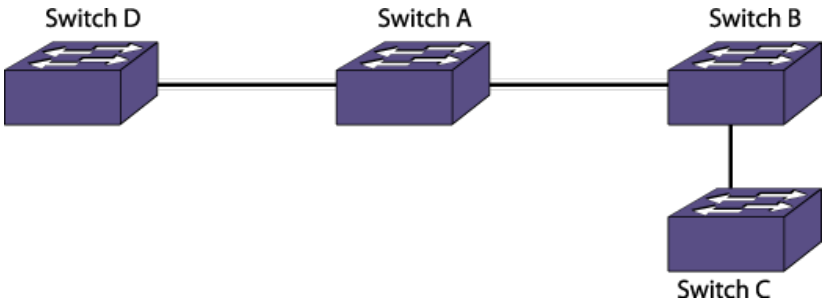


Figure 34: Route Aggregation Example

- Configure router A.

```
Configure router A.
create vlan "v3"
create vlan "v1"
create vlan "v2"
configure vlan v3 add ports 1
configure vlan v1 add ports 23
configure vlan v2 add ports 16
configure v1 ipaddress 10.1.1.1/24
configure v2 ipaddress 10.1.2.1/24
configure v3 ipaddress 10.1.4.1/24
enable ipforwarding
configure bgp AS-number 100
configure bgp routerid 1.1.1.1
create bgp neighbor 10.1.1.2 remote-AS-number 64500
create bgp neighbor 10.1.2.2 remote-AS-number 64505
create bgp neighbor 10.1.4.2 remote-AS-number 64510
enable bgp neighbor all
enable bgp
enable bgp aggregation
configure bgp add aggregate-address 172.16.0.0/16 as-set summary-only
```

- Configure router B.

```
Configure router B.
create vlan "v1"
create vlan "net"
configure vlan v1 add ports 5:33
configure vlan net add ports 5:20
configure v1 ipaddress 10.1.1.2/24
configure net ipaddress 172.16.1.1/24
enable ipforwarding
configure bgp AS-number 64500
configure bgp routerid 1.1.1.2
configure bgp add network address-family 172.16.1.0/24
create bgp neighbor 10.1.1.1 remote-AS-number 100
enable bgp neighbor 10.1.1.1
enable bgp
```

- Configure router C.

```
Configure router C.
create vlan "v1"
create vlan "net"
configure vlan net add ports 2:15
configure vlan v1 add ports 2:16
configure v1 ipaddress 10.1.2.2/24
configure net ipaddress 172.16.2.1/24
enable ipforwarding
configure bgp AS-number 64505
configure bgp routerid 2.1.1.2
configure bgp add network 172.16.2.0/24
```

```
create bgp neighbor 10.1.2.1 remote-AS-number 100
enable bgp neighbor 10.1.2.1
enable bgp
```

- Configure router D.

```
Configure router D.
create vlan "v1"
configure vlan v1 add ports 24
configure v1 ipaddress 10.1.4.2/24
enable ipforwarding
configure bgp AS-number 64510
configure bgp routerid 5.1.1.2
create bgp neighbor 10.1.4.1 remote-AS-number 100
enable bgp neighbor 10.1.4.1
enable bgp
```

- The following command displays the aggregated route at Router D:

```
* Switch.22 # show bgp routes all
Routes:
Destination         Peer            Next-Hop        LPref Weight MED        AS-Path
-------------------------------------------------------------------------------------
-
*>i  172.16.0.0/16         10.1.4.1        10.1.4.1        100   1      0          100
300
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 1
Feasible Routes   : 1
Active Routes     : 1
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 0
Routes from Ext Peer: 1
```

Route Aggregation Example for IPv6

- Configure router A.

```
create vlan "v3"
create vlan "v1"
create vlan "v2"
configure vlan v3 add ports 1
configure vlan v1 add ports 23
configure vlan v2 add ports 16
configure v1 ipaddress 2001:db8:1::1/48
configure v2 ipaddress 2001:db8:3::1/48
configure v3 ipaddress 2001:db8:5::1/48
enable ipforwarding ipv6
configure bgp AS-number 100
configure bgp routerid 1.1.1.1
create bgp neighbor 2001:db8:1::2 remote-AS-number 200
create bgp neighbor 2001:db8:3::2 remote-AS-number 300
create bgp neighbor 2001:db8:5::2 remote-AS-number 400
enable bgp neighbor all
enable bgp neighbor all capability ipv6-unicast
enable bgp
enable bgp aggregation
```

```
configure bgp add aggregate-address address-family ipv6-unicast 2001::/16 as-set
summary-only
```

- Configure router B.

```
create vlan "v1"
create vlan "net"
configure vlan v1 add ports 5:33
configure vlan net add ports 5:20
configure v1 ipaddress 2001:db8:1::2/48
configure net ipaddress 2001:db8:2222::1/48
enable ipforwarding ipv6
configure bgp AS-number 200
configure bgp routerid 1.1.1.2
configure bgp add network address-family ipv6-unicast 2001:db8:2222::/48
create bgp neighbor 2001:db8:1::1 remote-AS-number 100
enable bgp neighbor 2001:db8:1::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:1::1
enable bgp
```

- Configure router C.

```
create vlan "v1"
create vlan "net"
configure vlan net add ports 2:15
configure vlan v1 add ports 2:16
configure v1 ipaddress 2001:db8:3::2/48
configure net ipaddress 2001:db8:2333::2/48
enable ipforwarding ipv6
configure bgp AS-number 300
configure bgp routerid 2.1.1.2
configure bgp add network address-family ipv6-unicast 2001:db8:2333::/48
create bgp neighbor 2001:db8:3::1 remote-AS-number 100
enable bgp neighbor 2001:db8:3::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:3::1
enable bgp
```

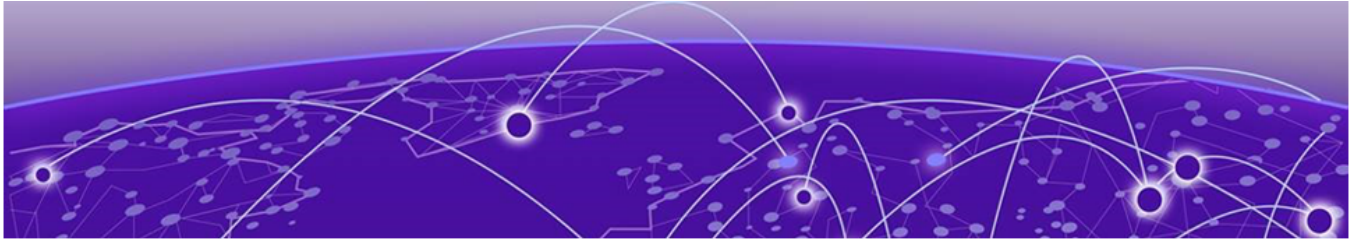
- Configure router D.

```
create vlan "v1"
configure vlan v1 add ports 24
configure v1 ipaddress 2001:db8:5::2/48
enable ipforwarding ipv6
configure bgp AS-number 400
configure bgp routerid 5.1.1.2
create bgp neighbor 2001:db8:5::1 remote-AS-number 100
enable bgp neighbor 2001:db8:5::1 capability ipv6-unicast
enable bgp neighbor 2001:db8:5::1
enable bgp
```

- The following command displays the aggregated route at Router D:

```
* switch # sh bgp routes address-family ipv6-unicast all
Routes:
Destination                                LPref Weight MED
Peer                                    Next-Hop
AS-Path
------------------------------------------------------------------------
*>i  2001::/16                                  100   1      0
2001:db8:5::1                           2001:db8:5::1
100 { 200 300 }
Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
(s) Stale, (m) Multipath, (u) Unfeasible
Origin: (?) Incomplete, (e) EGP, (i) IGP
BGP Route Statistics
Total Rxed Routes : 1
Feasible Routes   : 1
Active Routes     : 1
```

```
Rejected Routes   : 0
Unfeasible Routes : 0
Route Statistics on Session Type
Routes from Int Peer: 0
Routes from Ext Peer: 1
```



BGP Auto-peering

Auto-peering Introduction on page 484
BGP Auto-peering Feature Description on page 485
IP Multicast Forwarding (PIM-DM) on page 486
External Router Support on page 487
BGP Auto-peering Plug-and-Play Redundancy on page 488
Routing through the Underlay (VXLAN and AutoBGP Networks) on page 489
EVPN and VRF Support on page 491
Supported Platforms on page 494
BGP Auto-peering Feature Limitations on page 494
Configuring BGP Auto-peering on page 495
BGP Auto-peering Examples on page 496

Auto-peering Introduction

Auto-peering is a network of cooperating interconnected devices that create an AutoBGP for any topology, providing fully redundant, multipath routing. The fabric grows dynamically and freely, not bound to any well-known topology such as Clos or Leaf/Spine.

Auto-peering nodes build a secure network by running the very scalable Border Gateway Protocol (BGP) to exchange topology and host information about IP networks. It uses IPv6 as the network layer to transport IPv4 and IPv6 traffic.

Any device connecting to an auto-peering device is an attachment point to the network. This network provides the underlay for services such as *VXLAN (Virtual Extensible LAN)*, policy, VRF, and Fabric Attach. The AutoBGP device applies policy rules as it discovers external devices. These devices can be any IP host, LAG-attached servers and bridges, or gateway routers. By default, auto-peering allows connectivity for all attached hosts, allowing for a controller-less operation. However, interconnection (trunk ports) between fabric nodes should not be LAG ports; you should not enable port sharing on trunk ports.

Auto-peering uses a combination of the Link Layer Discovery Protocol (LLDP) and external BGP over an IPv6 Transport Layer to create the network. Auto-peering uses LLDP to discover surrounding nodes and the node's capabilities. LLDP is extended to advertise networking capabilities.

ExtremeXOS 31.1 introduces explicit CLI commands to configure OneConfig, which allows for ExtremeCloud IQ support for BGP Auto-peering (see Configuring BGP Auto-peering on page 495).

BGP Auto-peering Feature Description

Note
BGP auto-peering relies on a default STPD instance "s0" to resolve loops on the default VLAN.

Note
Auto-peering automatically creates VLANs dynamically when using the `create auto-peering` **`bgp routerid`** *`ipaddress`* **`AS-number`** *`asNumber`* command.

Note
The command `create auto-peering` **`bgp routerid`** *`ipaddress`* **`AS-number`** *`asNumber`* uses eBGP adjacencies. The AS number should be different for all devices participating in the network.

After executing the command `create auto-peering` **`bgp routerid`** *`ipaddress`* **`AS-number`** *`asNumber`* , the following series of events occurs, setting up the auto-peering:

1. AutoBGP capability is advertised out all ports by LLDP automatically. AutoBGP gets its configuration using cloud connector through Zero Touch Provisioning. A Python script initiates default VLAN and BGP configuration that is needed for AutoBGP.
2. When an AutoBGP-capable neighbor is detected out a port, LLDP notifies the AutoBGP manager.
3. The AutoBGP manager assigns the LLDP-learned port to one of the VLANs within its configured range and notifies VLAN manager. DHCP Relay is enabled on this VLAN.
4. AutoBGP informs LLDP of the IPv6 link-local address configured on that VLAN along with the BGP router ID and AS number.
5. LLDP sends the AutoBGP information (IPv6 link-local, router ID, and AS number) to the remote neighbor and learns the remote neighbor information and then send this to the AutoBGP manager.
6. The AutoBGP manager creates an AutoBGP route to the learned remote router ID in route manager.
7. The AutoBGP manager creates and enables the BGP neighbor.
8. The AutoBGP is formed and host route updates occur allowing end-to-end connectivity.
9. MBGP capability to carry VXLAN information is automatically enabled on each BGP peer when AutoBGP is enabled. In addition BGP is registered as a client of the Overlay Tunnel Manager (OTM), which maintains VNI/LTEP information.
10. The MBGP IPv4-VXLAN capability is used to advertise the Assisted Replication role if it is configured as "Replicator" or "Leaf".

11. The MBGP L2VPN-EVPN capability is automatically enabled to allow auto-peers to advertise and accept EVPN routes.
12. VXLAN tunnels are automatically programmed into OTM when EVPN routes are received containing a VNI and RTEP that is the BGP next-hop.

IP Multicast Forwarding (PIM-DM)

IP Multicast traffic within the BGP auto-peering network is forwarded using PIM-Dense Mode (PIM-DM) configured on ExtremeXOS devices. BGP auto-peering automatically enables PIM-DM globally for IPv4 and IPv6, as well as on each VLAN interface. Access VLANS must be configured and IPMC forwarding should be enabled to enable PIM-DM.

IPv4 Multicast sources and receivers from the access side are switched in the cloud. AutoBGP VLANs have only v6 interfaces, so PIM-DM is enabled in v6 mode on the network side. IPv6 network interfaces are added to the egress list of IPv4 cache entries so that IPv4 traffic reaches auto-peering devices. PIM-DM works on flood and prune mechanism so interfaces that have v6 and v4 neighbors or IGMP group joins receive the multicast traffic. IPv6 PIM-enabled interfaces process the IPv4 Multicast traffic and replicate it to the Multicast-enabled outgoing interfaces.

All the PIM-DM functionalities are supported in the auto-peering cloud:

- IPv4 cache miss on IPv6 interfaces
- Switching IPv4 Multicast traffic to IPv6 auto-peering interfaces
- MLD snooping
- PIM-DM join/prune, assert, graft, graft ACK
- PIM-DM state refresh
- ECMP (load splitting)

Auto-peering Connection with PIM-Spare Mode (SM) Domain

PIM-DM is able to connect with PIM-SM domains present outside of the auto-peering network. ExtremeXOS creates (*,*,RP) cache entries to pull the traffic from the SM domain to the auto-peering domain. The auto-peering network then applies dense behavior to the incoming Multicast traffic. Traffic originating from Dense Mode is carried until it meets a Multicast border router (MBR) node where both auto-peering and PIM-SM interfaces are present. PIM-SM becomes the first-hop router (FHR) for the incoming stream and generates a register message to RP.

Auto-peering Connection with PIM-DM Domain

PIM-DM interfaces switch and receive Multicast streams with non-auto-peering DM interfaces.

Existing PIM-SM Domains within the Auto-peering Network

SM and DM domains cannot overlap within the auto-peering network. Since auto-peering interfaces are configured with DM, existing SM configuration may not function as expected.

Limitations

- Slow path forwarding of IP Multicast traffic within an BGP auto-peering network is not supported.
- Avoiding “loss of first packet” of IP Multicast is not supported.
- PIM should be enabled on all the auto-peering devices, so PIM snooping is not recommended.
- PIM-SM and SSM modes are not supported within the auto-peering network.

External Router Support

A leaf node may support *OSPF (Open Shortest Path First)*v2 and *BGP (Border Gateway Protocol)* to function as a border leaf gateway. This node must be configured using the CLI to ensure the correct router ID and AS number is applied so that it interoperates with neighboring BGP or OSPFv2 non-auto-peering nodes.

The protocols Intermediate System to Intermediate System (ISIS) and *RIP (Routing Information Protocol)* are not supported.

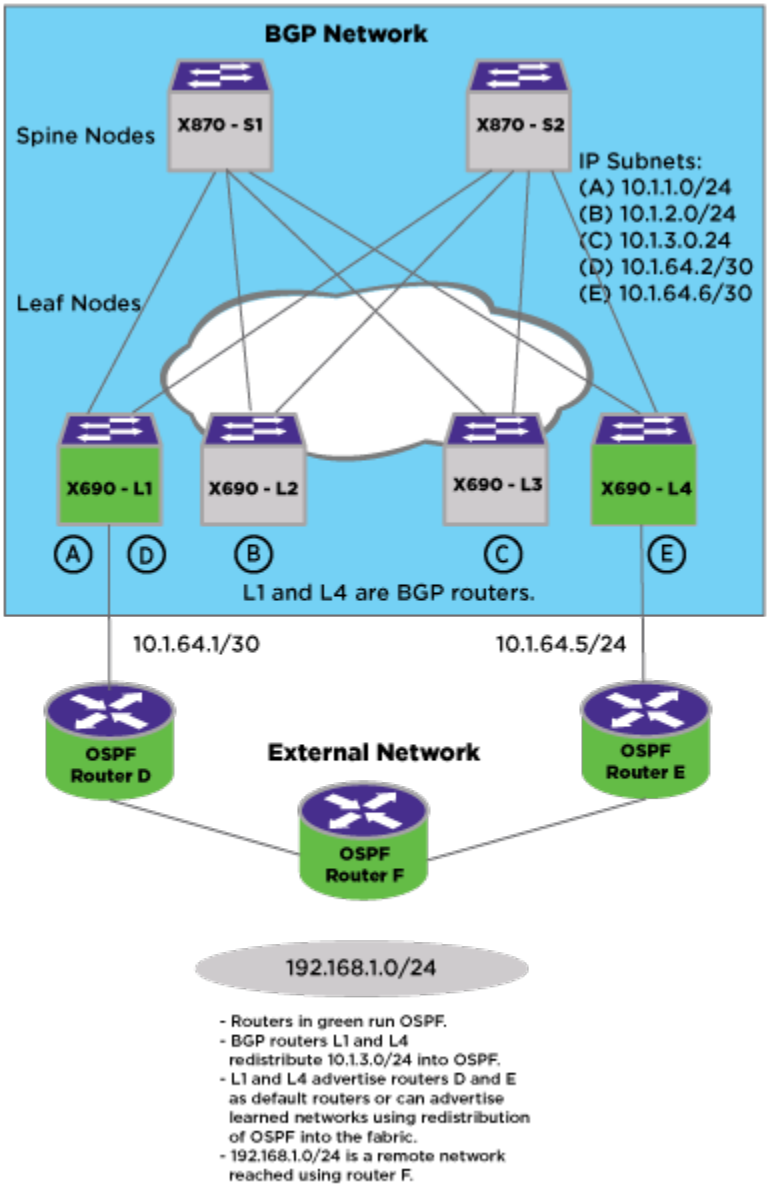


Figure 35: BGP Auto-peering with OSPF

BGP Auto-peering Plug-and-Play Redundancy

BGP auto-peering allows for devices to redundantly connect with network devices in any manner. EasyLAG, an active/standby service, is the default redundancy feature, which allows a LAG-connected device to attach to the network (see AutoBGP LAG Active/Standby Mode on page 489).

AutoBGP LAG Active/Standby Mode

For each Link Aggregation Control Protocol (LACP)-attached port, information on the port is distributed through the auto-peering network using a proprietary Border Gateway Protocol (BGP)-based mechanism. It uses the private AFI/SAFI to propagate the Ethernet segment ID and remote router ID as next hop.

This information is used, with a deterministic mechanism (lowest router ID), by a switch to place its port in standby mode. Standby mode means "Oper Status Down," and link down. If the active port goes down, the switch in the standby mode detects this using the same BGP mechanism, and makes its port active. Packets are reflected until BGP convergence of Ethernet segment ID to participating nodes.

For each attached client, only one attached port per switch is allowed. By default, after enabling auto-peering, each switch automatically supports creation of LAGs on all edge (client) ports, and prevents creation of LAGs on all auto-peering ports.

After BGP auto-peering is enabled on a switch (`create auto-peering` **`bgp routerid`** *`ipaddress`* **`AS-number`** *`asNumber`*), the equivalent of the following configuration is enacted for every Ethernet port (X) on the switch:

```
enable sharing X grouping X lacp
configure sharing X lacp fallback enable
configure sharing X lacp fallback timeout 1
```

This allows a LAG to come up on LAG-capable attached devices, and to have the port become a forwarding bridge port in one second if attached to a non-LAG capable (or LAG is disabled) device. If the attached device later becomes able to support LAG, LAG automatically comes up. In the same manner, as manually executing the commands, this produces a persistent state, and the commands appear in the configuration. This configuration persists after reboots.

If auto-peering is disabled on a switch, the equivalent of the following command is automatically executed:

```
disable sharing X
```

Routing through the Underlay (VXLAN and AutoBGP Networks)

It is valuable to provide a shortcut IPv4 routing service directly over the underlay and avoiding using an external router to perform the same function, and thereby reducing traffic “trombone” effect.

The following are possible scenarios:

- Attachment devices such as routers, firewalls, and servers that are connected on *VLAN (Virtual LAN)*s without a *VXLAN* enabled.

  Currently, the ingress fabric router responds to all *ARP (Address Resolution Protocol)*s regardless of target IP address. This is sometimes called “local proxy ARP” and is disabled on VXLAN interfaces. Local proxy is a valid setting with non-VXLAN VLANs and fabric routing.
- IP hosts attached to a VLAN associated with a VXLAN and having a VNI. They are directly routed at the ingress switch with no VTEP encapsulation if the packet has a

DMAC equal to VMAC as explained below and the destination host is in the routing table.

Normally, a host uses ARP for its default gateway when it must reach a destination that does not match its own network/subnet. A receiving auto-peering router responds with its virtual MAC for any ARP request whose target IP address matches its configured IP address, and if the source IP address is of the same subnet for which it is connected. Subsequent host packets destined to a remote host have a destination MAC address matching the VMAC and are routed directly at the ingress switch without VTEP encapsulation. Packets may be routed locally if any destination host is attached to the same ingress switch, but have a different VNI or in the case where they are connected on VLANs without VXLAN enabled. It is presumed that forwarding routers in the path have reachability to the destination host as they are learned using the *VRRP (Virtual Router Redundancy Protocol)* host mobility feature (see VRRP Host Mobility on page 2289) and propagated by *BGP* throughout the network. Any ARP packets not requesting the VIP of the auto-peering node are consumed on the ingress switch by ARP/L2 *FDB (forwarding database)* and the underlying VXLAN feature. Hence, subsequent packets with non-VMAC match FDB entries and those destinations are sent over VXLAN tunnels.

To allow the feature, configure an IP address and subnet mask and enable IP forwarding on a VLAN associated with a VNI. Enabling IPv6 forwarding should be not permitted, so IPv6 traffic must be tunneled. For ExtermeXOS 22.4 and later, along with the IPv6 restriction, this service is limited to the default VR. If VRF is required, do not use this feature. Additionally, VRRP and host mobility should also be enabled on the VLAN.

VXLAN to Non-VXLAN Routing in a Non-AutoBGP Network

ExtremeXOS 22.4 and later allow the IP Forwarding flag to be set for *VLAN* interfaces with a *VXLAN* association for IPv4 traffic. This qualifies any ExtremeSwitching switch that supports VXLAN but does not support routing in-and-out of tunnels (RIOT) (see Routing in and out of Tunnels (RIOT) on page 2311) or IP fabric.

For example, an ExtremeSwitching switch may use VXLAN over an *OSPF*-based underlay. It is possible a remote destination network is learned by OSPF on one of its underlay VLAN interfaces that has no VXLAN association. While the CLI allows this setup, this is not a validated design and is therefore not recommended.

VXLAN to Non-VXLAN Routing in an AutoBGP Network

Consider two pair of ExtremeSwitching switches as leaf nodes connected to two different pair of ExtremeSwitching switches as spine nodes with BGP auto-peering enabled. There is a mix of *VXLAN* and non-VXLAN *VLAN*s. In this case, all routing is accomplished through VLANs associated with VXLAN having VNIs or through other valid normal routing means and distributed to all auto-peering nodes. This configuration is valid and supported. VRRP and host mobility should be enabled on both VXLAN tenant and non-VXLAN VLANs for this to work.

VXLAN to VXLAN Routing in an AutoBGP Network

Consider hosts A and B that reside within an *BGP* auto-peering domain. Hosts A and B are attached on two different *VXLAN*s: A on VNI 1000 and B on VNI 2000. A connects directly to ExtremeSwitching -leaf1 on VLAN 10 while B connects to ExtremeSwitching -leaf2 on VLAN 20. The switches are connected to two spine switches of a different model using FBRC VLANs 100 and 200, respectively. In this case, all routing is accomplished through VLANs associated with VXLANs having VNIs. This configuration is valid and supported. In ExtremeXOS 22.5 and later, support routing in-and-out of tunnels (RIOT) (see Routing in and out of Tunnels (RIOT) on page 2311) is supported, so underlay routing is not needed in this case.

EVPN and VRF Support

BGP with EVPN carries virtual network identifier (VNI) and source virtual tunnel end point (VTEP) as part of the MP-BGP standard messaging payload, thereby enabling control plane learning of L2 and L3 reachability information for greater scale of *VXLAN* networks. As MAC addresses are learned, they are inserted into the EVPN instance table to be propagated by MP-BGP to all other VTEPS in AutoBGP. Learned MACs addresses using BGP are inserted into the local *FDB* and *ARP* tables associated with the virtual network. This provides robustness to VXLAN networks with a reduced reliance on flooding when combined with ARP suppression. For a description of EVPN control plane protocol, see *RFC 7432*. For a description of VXLAN and network virtualization overlay solution using EVPN, see *RFC draft-ietf-bess-overlay-08*.

EVPN Route Distinguisher and Route Target

A default route distinguisher (RD) per MAC address VRF is auto-assigned per provider edge (PE) as described in Section 7.9 of *RFC 7432*, Type 1 RD format is the PE loopback address, plus 12-bit Ethernet tag—one MAC VRF per Ethernet tag. Auto-derived route target (RT) is type 0 RD formatted as 2 byte AS number, plus 24-bit network services identifier (NSI) as described in Section 5.2.1 of *RFC draft-ietf-bess-overlay-08*. The default 2-byte AS number is 64,716. The RD is of local significance only—every VLAN defined on the node has the same number of MAC VRFs associated with it. For example, two AutoBGP nodes 1.1.1.1 and 2.2.2.2 with each having two untagged VLANS 10 and 20. When AutoBGP node 1.1.1.1 advertises EVPN NLRIs from two MAC VRFs using RDs of 1.1.1.1...10 and 1.1.1.1..20, it is possible that VLAN 10 on 1.1.1.1 is associated with VNI 1000 and on 2.2.2.2 VNI, 1000 is associated with VLAN 20. It is the NSI/VNI that brings the VXLAN together on the overlay and ensures that it is properly represented by the RT.

IP VRF Limitations

The following limitations apply to IP VRF:

- IP VRF route leaking is not supported.
- Route reflection is not supported.
- VRF is not supported for underlay routing.
- Static routes are supported per VRF, but only for external routers that are attached on a tenant VLAN.

Asymmetric Routing

AutoBGP nodes perform asymmetric routing for EVPN type 2 IP/MAC routes that are associated with a local virtual network identifier (VNI). They are installed into the *FDB* and *ARP* tables. They are not installed in the VRF RTM. Locally learned IP hosts using *VRRP* host-mobility may be installed in the VRF RTM. Asymmetric routing requires all VNIs to be created and associated with a VRF on all participating nodes.

Consider an AutoBGP network that supports an “Engineering” VRF consisting of two subnetted VNIs for hosts A through H, and VRFs are created with VNI mappings, using policy or other means. A host is placed into a VLAN on an attachment port of a device which corresponds to a VNI. In order to support asymmetric routing, all virtual networks are created on all nodes even if are no local host IP/MACs are discovered. If there is no VLAN assigned to a VNI, one is chosen and assigned an internal tag. The VLAN is given a loopback designation. If a host is later discovered, that is associated with another VLAN with same the VNI, the new VLAN replaces it and will have no loopback designation. Here, A on L1 routes to H on L4 as A-H are on two different VNIs. There are no local hosts on VNI 20 on L1, in which case, the loopback designation for the internally chosen VLAN enables asymmetric routing to take place. L4 then bridges to H. Troubleshooting can be achieved by looking into the FDBs and ARP tables on each node.

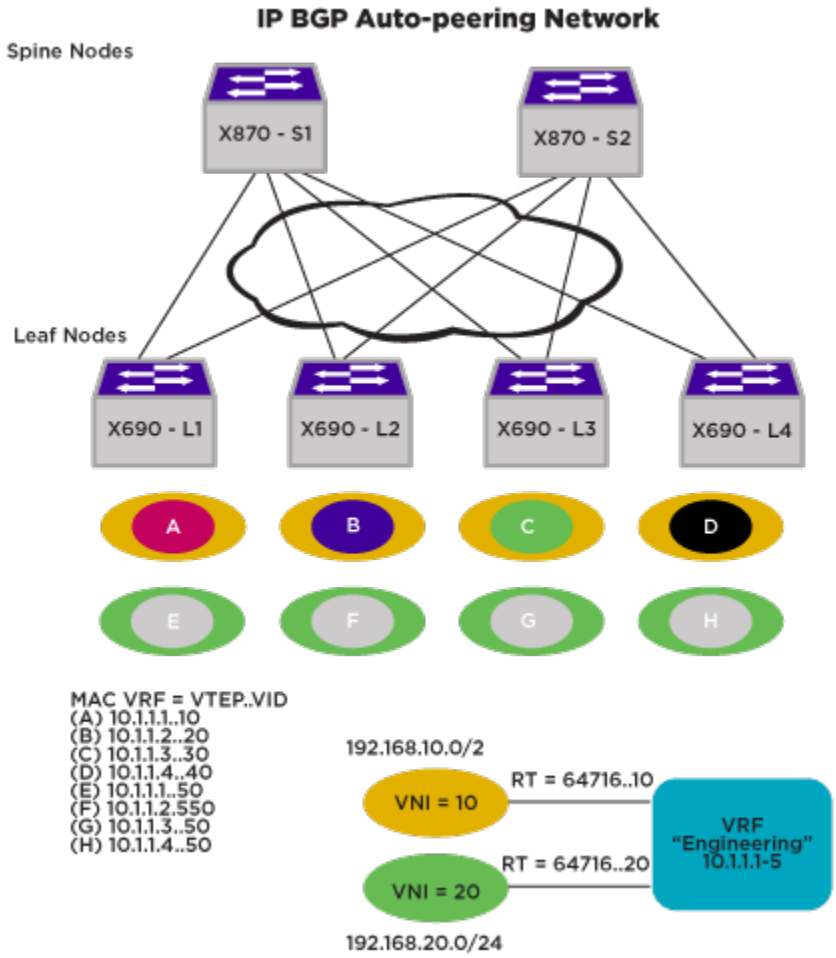


Figure 36: EVPN and Asymmetric VRF Routing in BGP Auto-peering

Consider hosts F and G that represent routers to external networks. Default static routes for hosts F and G are found as route entries in all nodes reachable on the local VLAN representing the VNI.

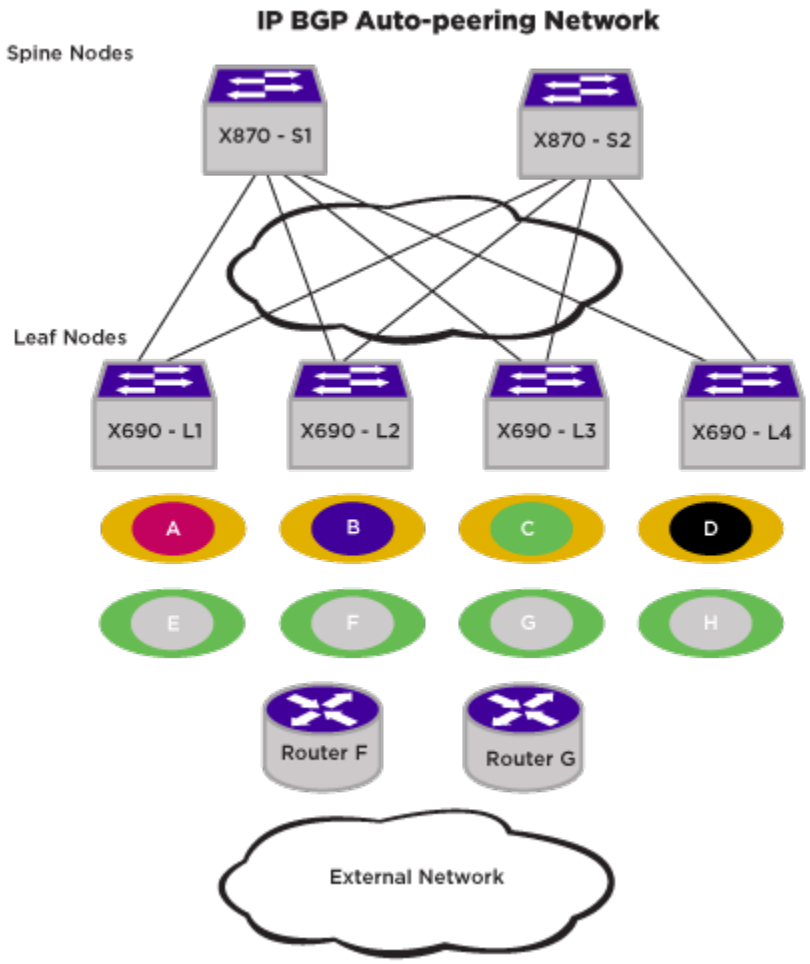


Figure 37: EVPN and Asymmetric VRF Routing in BGP Auto-peering with Routers

Supported Platforms

MSDP (Multicast Source Discovery Protocol) is supported on all platforms running a minimum software version of ExtremeXOS 12.0 with the Premier or Extreme Platform ONE license.

Our implementation of MSDP is compliant with RFC 3618 and RFC 3446, and compatible with other devices that are compliant with these standards.

BGP Auto-peering Feature Limitations

Note
BGP and OSPFv2 Auto-peering are mutually exclusive. Only one form of Auto-peering can exist at a time.

The following features are not supported in BGP auto-peering:

- Stacking
- AutoBGP LAG with Extended Edge Switching
- Static LAG attachments to AutoBGP
- AutoBGP links on Extended Edge Switching ports
- MPLS, VPLS, L2VPN, L3VPN
- PIM Snooping, PIM-SM, SSM mode
- OSPF, OSPFv3, ISIS, RIP, RIPNG per VRF
- IPv6 within VRF
- VRF route leaking
- EVPN multi-homing Ethernet segments
- Explicit-remotes mode used for VNET flooding
- AutoBGP MLAG
- EVPN does not interoperate with third-party devices

The following limitations apply:

- Manual configuration of MLAG is supported. One MLAG peer per leaf node.
- Loss of first multicast packet in the flow is expected due to slow path forwarding.
- Static router must be an external router per VRF.
- VLANS spanning multiple bridges, where each bridge is AutoBGP LAG connected, must be VXLAN-based, or AutoBGP LAG replaced with MLAG.
- VLANs behind a bridge that is AutoBGP LAG connected *and* AutoBGP node has same VLAN ports must be VXLAN-based or AutoBGP LAG replaced with MLAG.

Configuring BGP Auto-peering

To create and enable BGP auto-peering, use the command:

```
create auto-peering bgp routerid ipaddress AS-number asNumber
```

To delete BGP auto-peering, use the command:

```
delete auto-peering
```

To show the status of BGP auto-peering and the learned auto-peering interfaces and corresponding remote peer information, use the command:

```
show auto-peering {bgp | ospf}
```

OneConfig Commands

To add dynamic BOOTP relay servers that the DHCP relay agent uses to forward DHCP traffic received from host attachments, use the following command:

```
configure auto-peering one-config bootprelay [add | delete] [ip_address
| ipv6_address] vr vrname
```

To configure a list of OneConfig dynamic Auto-peering static routes, use the following command:

```
configure auto-peering one-config iproute [add | delete] [host | hostv6]
[[ipaddress {netmask} | ipNetmask] gateway | ipNetmaskv6 gatewayv6] {vr
vrname }
```

To specify a list of OneConfig dynamic auto-peering services, use the following command:

```
configure auto-peering one-config nsi-id id type [nsi | vrf] [add |
delete] [[ipaddress {netmask} | ipNetmask ] | ipNetmaskv6] {vr vrname }
```

To configure a list of unique values that identify the remote Auto-peering devices to which this device can also automatically form an adjacency, use the following command:

```
configure auto-peering one-config remote id [add | delete] id {password
[none | {encrypted} tcpPassword]}
```

To configure the auto-peering TCP MD5 password devices will configure in the TCP MD5, use the following command:

```
configure auto-peering one-config password [none | {encrypted}
tcpPassword]
```

To configure the ID used by each device when automatically forming an adjacency with an BGP Auto-peering neighbor, use the following command:

```
configure auto-peering one-config id [none | id]
```

To show auto-peered OneConfig information configured on the device, use the command:

```
show auto-peering one-config {remote | service | route | database |
bootprelay}
```

BGP Auto-peering Examples

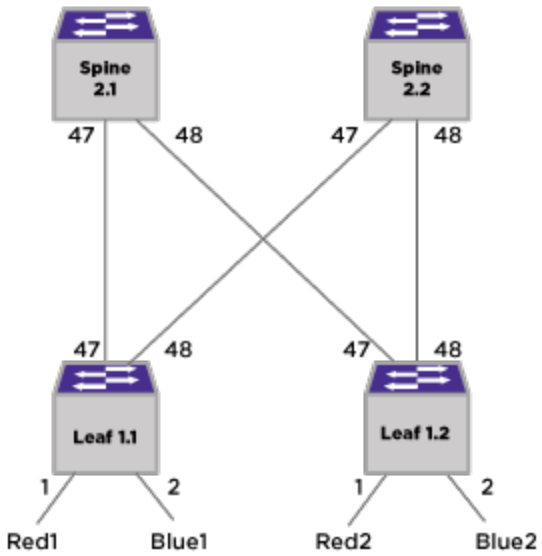
The following are some examples of BGP auto-peering configurations.

BGP Auto-peering Simple Leaf/Spine Example

The following example shows a simple leaf/spine topology with users on *VLAN*s, red and blue, noted as 1 and 2.

Note
An Advanced Edge or Base license must be installed on all devices.

Devices labeled with router IDs 10.255.x.y:



Leaf 1

```
configure vlan default delete ports 1-2
create vlan "red" tag 100
configure vlan red add ports 1 untagged
configure vlan "red" ipaddress 10.1.1.1/24
enable ipforwarding vlan "red"

create vrrp vlan "red" vrid 1
configure vrrp vlan "red" vrid 1 priority 255
configure vrrp vlan "red" vrid 1 host-mobility on
configure vrrp vlan "red" vrid 1 add 10.1.1.254
enable vrrp vlan "red" vrid 1

create vlan "blue" tag 200
configure vlan "blue" ipaddress 10.1.2.1/24
configure vlan blue add ports 2 untagged
enable ipforwarding vlan "blue"

create vrrp vlan "blue" vrid 1
configure vrrp vlan "blue" vrid 1 priority 255
configure vrrp vlan "blue" vrid 1 host-mobility on
configure vrrp vlan "blue" vrid 1 add 10.1.2.254
enable vrrp vlan "blue" vrid 1

create auto-peering bgp vlans routerid 10.255.1.1 AS-number 4278124801
```

Leaf 2

```
configure vlan default delete ports 1-2
create vlan "red" tag 100
configure vlan red add ports 1 untagged
configure vlan "red" ipaddress 10.1.1.1/24
enable ipforwarding vlan "red"

create vrrp vlan "red" vrid 1
```

```
configure vrrp vlan "red" vrid 1 priority 255
configure vrrp vlan "red" vrid 1 host-mobility on
configure vrrp vlan "red" vrid 1 add 10.1.1.1
enable vrrp vlan "red" vrid 1

create vlan “blue" tag 200
configure vlan "blue" ipaddress 10.1.2.1/24
configure vlan blue add ports 2 untagged
enable ipforwarding vlan "blue"

create vrrp vlan “blue" vrid 1
configure vrrp vlan "blue" vrid 1 priority 255
configure vrrp vlan "blue" vrid 1 host-mobility on
configure vrrp vlan "blue" vrid 1 add 10.1.2.1
enable vrrp vlan "blue" vrid 1
create auto-peering bgp routerid 10.255.1.2 AS-number 4278124802
```

Spine 1

```
# create auto-peering bgp routerid 10.255.2.1 AS-number 4278125057
# show auto-peering

ISSU     : In Service
Router ID: 10.255.2.1
AS       : 4278125057

VLAN            Neighbor IP Address           Router ID       Remote AS
--------------- ----------------------------- --------------- ----------
SYS_BGP_0001   fe80::204:96ff:fe97:fcd5      10.1.1.1        4278124801
SYS_BGP_0002   fe80::204:96ff:fe97:fdd5      10.1.1.2        4278124802
. . .
FBRC_VLAN_2064  None
```

Spine 2

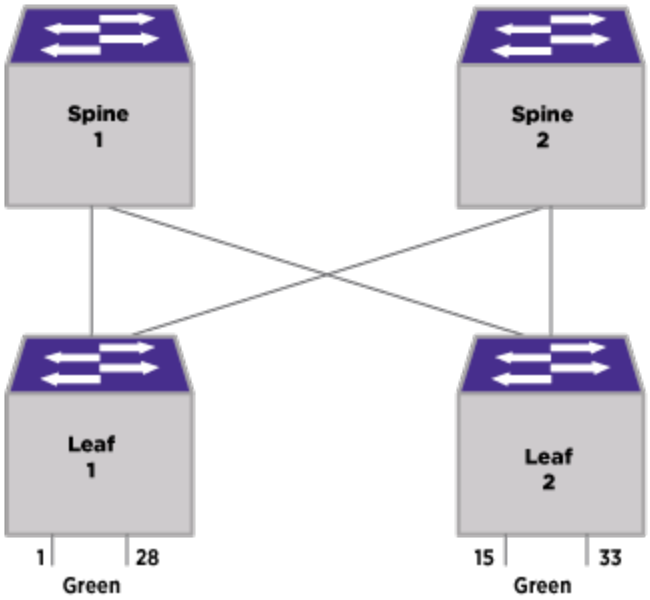
```
# create auto-peering bgp routerid 10.255.2.2 AS-number 4278125058
# show auto-peering

ISSU     : In Service
Router ID: 10.255.2.2
AS       : 4278125058

VLAN            Neighbor IP Address           Router ID       Remote AS
--------------- ----------------------------- --------------- ----------
SYS_BGP_0001  fe80::204:96ff:fe97:fcd5      10.1.1.1        4278124801
SYS_BGP_0002  fe80::204:96ff:fe97:fdd5      10.1.1.2        4278124802
. . .
FBRC_VLAN_2064  None
```

BGP Auto-peering Leaf/Spine with VXLAN Example

The following example shows a leaf/spine topology of ExtremeSwitching switches with *VXLAN* with users on *VLAN* green.

Note
An Advanced Edge or Base license must be installed on all switches.

The following commands reserve 10 VLANs.



Leaf 1

```
create auto-peering bgp routerid 1.1.1.120 AS-number 345200

create vlan "tenant"
configure vlan tenant tag 901
configure vlan tenant add ports 1,28 tagged
configure vlan tenant suppress arp-only (recommended but not necessary)

configure vlan tenant ipaddress 10.1.1.1 255.255.255.0
disable igmp snooping vlan "tenant"
enable ipforwarding vlan tenant

create vrrp vlan tenant vrid 151
configure vrrp vlan tenant vrid 151 host-mobility on
configure vrrp vlan tenant vrid 151 add 10.1.1.254
enable vrrp vlan tenant vrid 151

create virtual-network "vni1" flooding standard
configure virtual-network local-endpoint ipaddress 1.1.1.120
configure virtual-network "vni1" vxlan vni 1
configure virtual-network "vni1" add vlan tenant
```

Leaf 2

```
create auto-peering bgp routerid  1.1.1.121 AS-number 123500

create vlan "tenant"
configure vlan tenant tag 901
configure vlan tenant add ports 15,33 tagged
configure vlan tenant ipaddress 10.1.1.1 255.255.255.0
configure vlan tenant suppress arp-only (recommended but not necessary)
disable igmp snooping vlan "tenant"
enable ipforwarding vlan tenant

create vrrp vlan tenant vrid 151
```

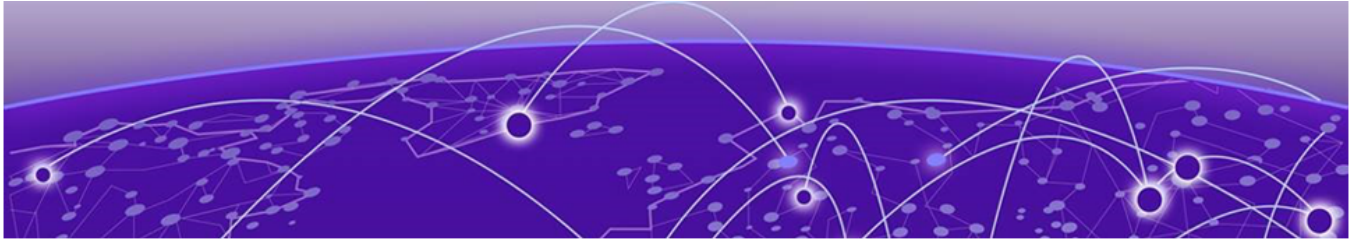
```
configure vrrp vlan tenant vrid 151 host-mobility on
configure vrrp vlan tenant vrid 151 add 10.1.1.254
enable vrrp vlan tenant vrid 151

configure virtual-network "vni1" add vlan tenant
configure virtual-network local-endpoint ipaddress 1.1.1.121 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" vxlan vni 1
configure virtual-network "vni1" add vlan tenant
```

Spine 1

```
# create auto-peering bgp routerid 10.255.2.1 AS-number 4278125057
# show auto-peering

ISSU     : In Service
Router ID: 10.255.2.1
AS       : 4278125057

VLAN            Neighbor IP Address           Router ID       Remote AS
--------------- ----------------------------- --------------- ----------
SYS_BGP_0001    fe80::204:96ff:fe97:fcd5       14.33.13.2       345201
SYS_BGP_0002    fe80::204:96ff:fe97:fdd5       12.22.2.120      345200

. . .
FBRC_VLAN_2064  None
```

Spine 2

```
# create auto-peering bgp routerid 10.255.2.2 AS-number 4278125058
# show auto-peering

ISSU     : In Service
Router ID: 10.255.2.2
AS       : 4278125058

VLAN            Neighbor IP Address           Router ID       Remote AS
--------------- ----------------------------- --------------- ----------
SYS_BGP_0001    fe80::204:96ff:fe97:fcd5       14.33.13.2       345201
SYS_BGP_0002    fe80::204:96ff:fe97:fdd5       12.22.2.120      345200
. . .
FBRC_VLAN_2064  None
```



CLEAR-Flow

CLEAR-Flow Overview on page 501
Adding CLEAR-Flow Rules to ACLs on page 502
Configuring CLEAR-Flow on page 515
Displaying CLEAR-Flow Configuration and Activity on page 516
CLEAR-Flow Rule Examples on page 516

This chapter offers detailed information about the software's implementation of CLEAR-Flow. This section provides an overview, as well as specific information on how to configure CLEAR-Flow, add CLEAR-Flow rules, and provides examples.

CLEAR-Flow Overview

CLEAR-Flow is a broad framework for implementing security, monitoring, and anomaly detection in ExtremeXOS software. Instead of simply looking at the source and destination of traffic, CLEAR-Flow allows you to specify certain types of traffic that require more attention. After certain criteria for this traffic are met, the switch can either take an immediate, predetermined action, or send a copy of the traffic off-switch for analysis.

CLEAR-Flow is an extension to Access Control Lists (ACLs). You create ACL policy rules to count packets of interest. CLEAR-Flow rules are added to the policy to monitor these ACL counter statistics. The CLEAR-Flow agent monitors the counters for the situations of interest to you and your network. You can monitor the cumulative value of a counter, the change to a counter over a sampling interval, the ratio of two counters, or even the ratio of the changes of two counters over an interval. For example, you can monitor the ratio between TCP SYN and TCP packets. An abnormally large ratio may indicate a SYN attack.

The counters used in CLEAR-Flow are either defined by you in an ACL entry, or can be a predefined counter. See Predefined CLEAR-Flow Counters on page 512 for a list and description of these counters.

If the rule conditions are met, the CLEAR-Flow actions configured in the rule are executed. The switch can respond by modifying an ACL that will block, prioritize, or

mirror the traffic, executing a set of CLI commands, or sending a report using a *SNMP (Simple Network Management Protocol)* trap or EMS log message.

Note

CLEAR-Flow is available on platforms with an Edge (or higher) or Base (or higher) license. For more license information, see the Switch Engine v33.6.1 Licensing Guide document.

CLEAR-Flow is supported only on ingress. Any limitations on a given platform for a regular ACL also hold true for CLEAR-Flow.

Adding CLEAR-Flow Rules to ACLs

As described in the *ACL (Access Control List)*s chapter, each ACL policy file consists of a number of named entries. Each entry consists of match conditions and actions to take if the entry is matched. CLEAR-Flow builds on the ACL concept to include rules that are periodically checked, and actions to take if a rule is triggered. The CLEAR-Flow entries are similar to the ACL entries.

The syntax of a CLEAR-Flow rule entry is:

```
entry <CLFrulename> {
if <match-type> { <match-conditions>;
}
Then {
<actions>;
}
}
```

Or you can specify an optional else clause:

```
entry <CLFrulename> {
if <match-type> { <match-conditions>;
}
Then {
<actions>;
} else {
<actions>;
}
}
```

In the CLEAR-Flow rule syntax, the *CLFrulename* is the name of the rule (maximum of 31 characters). The *match-type* specifies whether the rule is triggered when any of the expressions that make up the conditions are true (logical OR), or only when all of the expressions are true (logical AND). The *match-type* is an optional element. The *match-conditions* specifies the conditions that will trigger the rule, and how often to evaluate the rule. The *actions* in the then clause is the list of actions to take when the

rule is triggered, and the optional else clause *actions* is the list of actions to take after the rule is triggered, and when the *match-conditions* later become false.

Note
When you create an ACL policy file that contains CLEAR-Flow rules, the CLEAR-Flow rules do not have any precedence, unlike the ACL entries. Each CLEAR-Flow rule specifies how often it should be evaluated. The order of evaluation depends on the sampling time and when the CLEAR-Flow agent receives the counter statistics. The order of the CLEAR-Flow rules in the policy file does not have any significance.

CLEAR-Flow Rule Match Type

Match types are optional; the possible choices are:

- match all—All the match expressions must be true for a match to occur. This is the default.
- match any—If any match expression is true, then a match occurs.

CLEAR-Flow Rule Match Conditions

In a CLEAR-Flow rule, the *match-conditions* portion consists of one to four expressions, an optional global-rule statement, and an optional period statement:

```
entry <CLFrulename> {
if <match-type> { <expression>;
<expression>;
<expression>;
<expression>;
global-rule;
period <interval>;
}
Then {
<actions>;
} else {
<actions>;
}
}
```

In the following example, the CLEAR-Flow rule (named cflow_count_rule_example) will be evaluated every ten seconds. The actions statements will be triggered if the value of counter1 (defined earlier in the *ACL* policy file) is greater than 1,000,000:

```
entry cflow_count_rule_example  {
if { count counter1 > 1000000 ;
period 10 ;
}
Then {
<actions>;
}
}
```

The global-rule statement is optional and affects how the counters are treated. An ACL that defines counters can be applied to more than one interface. You can specify the global-rule statement so that counters are evaluated for all the applied interfaces. For example, if a policy that defines a counter is applied to port 1:1 and 2:1, a CLEAR-Flow rule that used the global-rule statement would sum up the counts from both ports. Without the global-rule statement, the CLEAR-Flow rule would look at only the counts received on one port at a time.

The period *interval* statement is optional and sets the sampling interval, in seconds. This statement specifies how often the rule is evaluated by the CLEAR-Flow agent. If not specified, the default value is 5 seconds.

The five CLEAR-Flow rule expressions are: count; delta; ratio; delta-ratio; and rule. All of these expressions check the values of counters to evaluate if an action should be taken. The counters are either defined in the ACL entries that are defined on the switch, or are the predefined CLEAR-Flow counters. When you use a counter statement in an ACL, you are defining the counter used by CLEAR-Flow to monitor your system.

Count Expression

A CLEAR-Flow count expression compares a counter with the threshold value. The following is the syntax for a CLEAR-Flow count expression:

```
count <counterName> REL_OPER <countThreshold> ;
hysteresis <hysteresis> ;
```

The value of *countThreshold* and *hysteresis* can be specified as floating point numbers. The count statement specifies how to compare a counter with its threshold. The *counterName* is the name of an *ACL* counter referred to by an ACL rule entry and the *countThreshold* is the value compared with the counter. The REL_OPER is selected from the relational operators for greater than, greater than or equal to, less than, or less than or equal to (>, >=, <, <=).

The hysteresis *hysteresis* statement is optional and sets a hysteresis value for the threshold. After the count statement is true, the value of the threshold is adjusted so that a change smaller than the hysteresis value will not cause the statement to become false. For statements using the REL_OPER > or >=, the hysteresis value is subtracted from the threshold; for < or <=, the hysteresis value is added to the threshold.

Following is an example of a count expression used in a CLEAR-Flow rule:

```
entry cflow_count_rule_example  {
if { count counter1 > 1000000 ;
period 10 ;
}
Then {
<actions>;
}
}
```

Actions are discussed in CLEAR-Flow Rule Actions on page 510.

Table 32 is an example of evaluating the CLEAR-Flow count expression above multiple times. Notice that the rule is not triggered until evaluation 3, when the value of the counter is greater than 1,000,000.

Table 32: Count Expression Evaluation Example

Evaluation	counter1 value	Rule triggered?
1	384625	No
2	769250	No
3	1153875	Yes
4	1538500	Yes

See Count Expression Example on page 516 for a full example of an ACL and a CLEAR-Flow rule using a count expression.

Delta Expression

A CLEAR-Flow delta expression computes the difference from one sample to the next of a counter value.

This difference is compared with the threshold value. The following is the syntax for a CLEAR-Flow delta expression:

```
delta <counterName> REL_OPER <countThreshold> ;
hysteresis <hysteresis> ;
```

The values of *countThreshold* and *hysteresis* can be specified as floating point numbers. The delta expression specifies how to compare the difference in a counter value from one sample to the next with its threshold. The *counterName* is the name of an *ACL* counter referred to by an ACL rule entry and the *countThreshold* is the value compared with the difference in the counter from one sample to the next. The REL_OPER is selected from the relational operators for greater than, greater than or equal to, less than, or less than or equal to (>, >=, <, <=).

The hysteresis *hysteresis* statement is optional and sets a hysteresis value for the threshold. After the delta statement is true, the value of the threshold is adjusted so that a change smaller than the hysteresis value will not cause the statement to become false. For statements using the REL_OPER > or >=, the hysteresis value is subtracted from the threshold; for < or <=, the hysteresis value is added to the threshold.

For example, the following delta expression:

```
delta counter1 >= 100 ;
hysteresis 10 ;
```

will only be true after the delta of the counter reaches at least 100. At the time it becomes true, the hysteresis value is subtracted from the threshold (setting the threshold to 90). With the threshold now at 90, the condition would stay true until the delta of the counter becomes less than 90.

If the expression becomes false, the threshold is reset to its original value. You would use the hysteresis value to prevent the expression from vacillating between the true and false states if the difference between the counter values is near the threshold. If the hysteresis value is greater than the threshold value, the hysteresis value will be set to 0.

Table 33 is an example of evaluating the CLEAR-Flow delta expression above multiple times. Notice that the rule is not triggered until evaluation 4, when the delta value (the change in the counter value from one evaluation to the next) is greater than or equal to 100. After the rule is triggered, it remains triggered until the delta value is less than 90 (the original threshold minus the hysteresis), at evaluation 7. At evaluation 9, the rule is again triggered when the delta reaches 100. The rule will remain triggered until the delta drops below 90.

Table 33: Delta Expression Evaluation Example

Evaluation	counter1 value	Delta value	Rule Triggered?
1	397	N/A	No
2	467	70	No
3	547	80	No
4	657	110	Yes
5	757	100	Yes
6	852	95	Yes
7	932	80	No
8	1031	99	No
9	1131	100	Yes
10	1230	99	Yes

See Delta Expression Example on page 517 for a full example of an ACL and a CLEAR-Flow rule using a delta expression.

Ratio Expression

A CLEAR-Flow ratio expression compares the ratio of two counter values with the threshold value.

The following is the syntax for a CLEAR-Flow ratio expression:

```
ratio <counterNameA> <counterNameB> REL_OPER <countThreshold> ;
min-value <min-value> ;
hysteresis <hysteresis> ;
```

The values of *countThreshold* and *hysteresis* can be specified as floating point numbers, and the ratio is computed as a floating point number. The ratio statement specifies how to compare the ratio of two counters with its threshold. To compute the ratio, the value of *counterNameA* is divided by the value of *counterNameB*. That ratio is then compared with the *countThreshold*. The REL_OPER is selected from the relational operators for greater than, greater than or equal to, less than, or less than or equal to (>, >=, <, <=).

The min-value statement is optional, and sets a minimum value for the counters. If either counter is less than the minimum value, the expression evaluates to false. If not specified, the minimum value is 1.

The hysteresis *hysteresis* statement is optional and sets a hysteresis value for the threshold. After the ratio statement is true, the value of the threshold is adjusted so that a change smaller than the hysteresis value will not cause the statement to become false. For statements using the REL_OPER > or >=, the hysteresis value is subtracted from the threshold; for < or <=, the hysteresis value is added to the threshold.

For example, the following ratio expression:

```
ratio counter1 counter2 >= 5 ;
min-value 100;
hysteresis 1 ;
```

is true only after the ratio of the counters reaches at least 5 and the counter values are at least 100. At the time it became true, the hysteresis value would be subtracted from the threshold (setting the threshold to 4). With the threshold now at 4, the condition would stay true until the ratio of the counters became less than 4.

If the statement becomes false, the threshold is reset to its original value. You can use the hysteresis value to prevent the rule from vacillating between the true and false states if the ratio between the counter values is near the threshold. If the hysteresis value is greater than the threshold value, the hysteresis value will be set to 0.

Table 34 is an example of evaluating the CLEAR-Flow ratio expression above multiple times. Notice that the rule is not triggered at the first evaluation because both counters have not yet reached the min-value of 100. The rule first triggers at evaluation 3, when ratio of the two counters exceeds 5. After the rule is triggered, it remains triggered until the ratio value is less than 4 (the original threshold minus the hysteresis), at evaluation 5. At evaluation 7, the rule is again triggered when the ratio reaches 5. The rule will remain triggered until the ratio drops below 4.

Table 34: Ratio Expression Evaluation Example

Evaluation	counter1 value	counter2 value	Ratio	Rule Triggered?
1	427	70	6	No
2	941	235	4	No
3	2475	412	6	Yes
4	2308	570	4	Yes
5	2313	771	3	No
6	3597	899	4	No
7	5340	1065	5	Yes

See Ratio Expression Example on page 518 for a full example of an *ACL* and a CLEAR-Flow rule using a ratio expression.

Delta-Ratio Expression

A CLEAR-Flow delta-ratio expression is a combination of the delta and ratio expressions.

The CLEAR-Flow agent computes the difference from one sample to the next for each of the two counters. The ratio of the differences is then compared to the threshold value. The following is the syntax for a CLEAR-Flow delta-ratio expression (note the similarity to the delta expression):

```
delta-ratio <counterNameA> <counterNameB> REL_OPER <countThreshold> ;
min-value <min-value> ;
hysteresis <hysteresis> ;
```

The values of *countThreshold* and *hysteresis* can be specified as floating point numbers, and the delta-ratio is computed as a floating point number. The delta-ratio statement specifies how to compare the ratio of the counter differences with its threshold. The difference of the sample values of *counterNameA* is divided by the difference of the sample values of *counterNameB*, to compute the ratio that is compared with the *countThreshold*. The REL_OPER is selected from the relational operators for greater than, greater than or equal to, less than, or less than or equal to (>, >=, <, <=).

The min-value statement is optional and sets a minimum value for the counters. If either counter is less than the minimum value, the expression evaluates to false. If not specified, the minimum value is 1.

The hysteresis *hysteresis* statement is optional, and sets a hysteresis value for the threshold. After the ratio statement is true, the value of the threshold is adjusted so that a change smaller than the hysteresis value will not cause the statement to become false. For statements using the REL_OPER > or >=, the hysteresis value is subtracted from the threshold; for < or <=, the hysteresis value is added to the threshold.

For example, the following delta-ratio expression:

```
delta-ratio counter1 counter2 >= 5 ;
min-value 100 ;
hysteresis 1 ;
```

will only be true after the ratio of the deltas of the counters reached at least 5. At the time it became true, the hysteresis value would be subtracted from the threshold (setting the threshold to 4). With the threshold now at 4, the condition would stay true until the ratio of the deltas of the counters became less than 4.

If the statement becomes false, the threshold is reset to its original value. You can use the hysteresis value to prevent the rule from vacillating between the true and false states if the ratio of the deltas of the counters is near the threshold. If the hysteresis value is greater than the threshold value, the hysteresis value will be set to 0.

Table 35 is an example of evaluating the CLEAR-Flow delta-ratio expression above multiple times. Notice that the rule is not triggered at the second evaluation because both counters have not yet reached the min-value of 100. The rule first triggers at evaluation 4, when ratio of the two counters exceeds 5. After the rule is triggered, it remains triggered until the ratio value is less than 4 (the original threshold minus the

hysteresis), at evaluation 6. At evaluation 8, the rule is again triggered when the ratio reaches 5. The rule will remain triggered until the ratio drops below 4.

Table 35: Delta-Ratio Expression Evaluation Example

Evaluation	counter1 value	counter1 delta	counter2 value	counter2 delta	Ratio	Rule Triggered?
1	110	N/A	20	N/A	N/A	No
2	537	427	90	70	6	No
3	1478	941	325	235	4	No
4	3953	2475	737	412	6	Yes
5	6261	2308	1307	570	4	Yes
6	8574	2313	2078	771	3	No
7	12171	3597	2977	899	4	No
8	17511	5340	4042	1065	5	Yes

See Delta-Ratio Expression Example on page 518 for a full example of an *ACL* and a CLEAR-Flow rule using a delta-ratio expression.

Rule-True-Count Expression

A CLEAR-Flow rule-true-count expression compares how many times a CLEAR-Flow rule is true with a threshold value.

One use is to combine multiple rules together into a complex rule. The following is the syntax for a CLEAR-Flow rule-true-count expression:

```
rule-true-count <ruleName> REL_OPER <countThreshold> ;
```

The rule-true-count statement specifies how to compare how many times a CLEAR-Flow rule is true with the expression threshold. The *ruleName* is the name of the CLEAR-Flow rule to monitor and the *countThreshold* is the value compared with the number of times the rule is true. The REL_OPER is selected from the relational operators for greater than, greater than or equal to, less than, or less than or equal to (>, >=, <, <=).

For example, the following delta-ratio expression:

```
rule-true-count cflow_count_rule_example >= 5 ;
```

will only be true after the CLEAR-Flow rule cflow_count_rule_example has been true at least five times. If the rule cflow_count_rule_example becomes true and remains true, and the period for cflow_count_rule_example is the default five seconds, the rule would have to be true for at least 20 seconds before the rule-true-count expression will become true. If the period of the rule cflow_count_rule_example is 10 seconds, it will need to be true for at least 40 seconds before the rule-true_count expression becomes true.

CLEAR-Flow Rule Actions

CLEAR-Flow rules specify an action to take when the rule is triggered and can optionally specify an action to take when the expression is false.

Because more than one action can be taken in a single rule, the collection of actions is referred to as an action list. The following sections describe the different rule actions:

- Permit/Deny on page 510
- QoS Profile on page 510
- Mirror on page 510
- SNMP Trap on page 511
- syslog on page 511
- CLI on page 511

Additionally, the *SNMP* trap, syslog, and CLI rule actions can use keyword substitution to make the rule actions more flexible. The keyword substitutions are described at the end of the rule action descriptions. See Keyword Substitution on page 511 for more information.

Permit/Deny

This action modifies an existing *ACL* rule to permit or block traffic that matches that rule.

- To change an ACL to permit, use the following syntax:

  ```
  permit <ACLRuleName>
  ```

- To change an ACL to deny, use the following syntax:

  ```
  deny <ACLRuleName>
  ```

QoS Profile

This action modifies an existing *ACL* rule to set the *QoS (Quality of Service)* profile for traffic that matches that rule.

- To change the ACL to forward to QoS profile <QPx>, use the following syntax:

  ```
  qosprofile <ACLRuleName> <QPx>
  ```

  ```
  For example:
  qosprofile acl_rule_1 QP3
  ```

Mirror

This action modifies an existing *ACL* rule to mirror traffic that matches that rule, or to stop mirroring that traffic. The mirroring port must be enabled when mirroring on an ACL rule is turned on. This could be configured earlier, or use the CLI action to execute CLI commands to configure mirroring at the same time.

- To change the ACL to mirror traffic, use the following syntax:

  ```
  mirror [add|delete] <ACLRuleName>
  ```

  For example (enabling mirroring from within CLEAR-Flow rule):

  ```
  enable mirror to port 7:4 tagged
  mirror add acl_rule_1
  ```

SNMP Trap

This action sends an *SNMP* trap message to the trap server, with a configurable ID and message string, when the rule is triggered. The message is sent periodically with interval *period* seconds. If *period* is 0, or if this optional parameter is not present, the message is sent only once when the rule is triggered. The interval must be a multiple of the rule sampling/evaluation interval, or the value will be rounded down to a multiple of the rule sampling/evaluation interval.

- To send an SNMP trap, use the following syntax:

  ```
  snmptrap <id> <message> <period>
  ```

syslog

A protocol used for the transmission of event notification messages across networks, originally developed on the University of California Berkeley Software Distribution (BSD) TCP/IP system implementations, and now embedded in many other operating systems and networked devices. A device generates a messages, a relay receives and forwards the messages, and a collector (a syslog server) receives the messages without relaying them.
syslog uses the UDP as its underlying transport layer mechanism. The UDP port that has been assigned to syslog is 514. (RFC 3164)

CLI

Command Line Interface. The CLI provides an environment to issue commands to monitor and manage switches and wireless appliances.

Keyword Substitution

To make the *SNMP* trap, syslog, and CLI actions more flexible, keyword substitutions are supported in the syslog and SNMP trap message strings, as well as in the CLI command strings.

Table 36 on page 511 lists the keywords and their substitutions.

If a keyword is not supported, or a counter name is not found, a string of "unknownKeyword[$keyword]" will be substituted.

For the $vlanName and $port keyword, the keyword **all** will be substituted for those rules in the wildcard *ACL* Some CLI commands do not support the **all** keyword, so caution must be used with CLI commands that use this feature.

A maximum of ten different counter substitutions can be used per rule, including counters used in expressions. For example, if a rule uses four counters in its expressions, then we can use six more different counters in keyword substitutions, for a total of ten.

Table 36: Keyword Substitutions

Keyword	Substitution
$policyName	Replace with the policy name.
$ruleName	Replace with the CLEAR-Flow rule name.
$<counterName>	Replace with counter value for the indicated counter name.

Table 36: Keyword Substitutions (continued)

Keyword	Substitution
$ruleValue	Replace with the current expression value.
$ruleThreshold	Replace with the expression threshold value.
$ruleInterval	Replace with the rule sampling/evaluation interval.
$vlanName	Replace with the interface *VLAN (Virtual LAN)* name.
$port	Replace with the interface port number.

Predefined CLEAR-Flow Counters

A number of packet statistics are gathered by the ExtremeXOS kernel.

To allow you to use these statistics in CLEAR-Flow expressions, these kernel counters are now available for use with CLEAR-Flow. Most of the counter names are based directly on well known names from common kernel structures and MIBs. The names are modified from their familiar form by pre-pending the characters sys_ to the counter names.

Table 37: Predefined CLEAR-Flow Counters

Counter Name	Description
sys_IpInReceives	The total number of input IP packets received from interfaces, including those received in error.
sys_IpInHdrErrors	The number of input IP packets discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.
sys_IpInAddrErrors	The number of input IP packets discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported Classes (for example, Class E).
sys_IpForwDatagrams	The number of input IP packets for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.
sys_IpInUnknownProtos	The number of locally-addressed IP packets received successfully but discarded because of an unknown or unsupported protocol.
sys_IpInDiscards	The number of input IP packets for which no problems were encountered to prevent their continued processing, but which were discarded (for example, for lack of buffer space). Note that this counter does not include any IP packets discarded while awaiting re-assembly.

[4] Most of these descriptions can be found in RFC 2011, SNMPv2 Management Information Base for the Internet Protocol using SMIv2.

Table 37: Predefined CLEAR-Flow Counters (continued)

Counter Name	Description
sys_IpInDelivers	The total number of input IP packets successfully delivered to IP user-protocols (including *ICMP (Internet Control Message Protocol)*).
sys_IpOutRequests	The total number of IP packets which local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any IP packets counted in ipForwDatagrams.
sys_IpOutDiscards	The number of output IP packets for which no problem was encountered to prevent their transmission to their destination, but which were discarded (for example, for lack of buffer space). Note that this counter would include IP packets counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.
sys_IpOutNoRoutes	The number of IP packets discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this `no-route' criterion.
sys_IpReasmTimeout	The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.
sys_IpReasmReqds	The number of IP fragments received which needed to be reassembled at this entity.
sys_IpReasmOKs	The number of IP packets successfully re-assembled.
sys_IpReasmFails	The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc.). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.
sys_IpFragOKs	The number of IP packets that have been successfully fragmented at this entity.
sys_IpFragFails	The number of IP packets that have been discarded because they needed to be fragmented at this entity but could not be, for example, because their Don't Fragment flag was set.
sys_IpFragCreates	The number of IP packet fragments that have been generated as a result of fragmentation at this entity.
sys_IcmpInMsgs	The total number of ICMP messages which the entity received. Note that this counter includes all those counted by icmpInErrors.
sys_IcmpInErrors	The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).
sys_IcmpInDestUnreachs	The number of ICMP Destination Unreachable messages received.
sys_IcmpInTimeExcds	The number of ICMP Time Exceeded messages received.
sys_IcmpInParmProbs	The number of ICMP Parameter Problem messages received.

Table 37: Predefined CLEAR-Flow Counters (continued)

Counter Name	Description
sys_IcmpInSrcQuenchs	The number of ICMP Source Quench messages received.
sys_IcmpInRedirects	The number of ICMP Redirect messages received.
sys_IcmpInEchos	The number of ICMP Echo (request) messages received.
sys_IcmpInEchoReps	The number of ICMP Echo Reply messages received.
sys_IcmpInTimestamps	The number of ICMP Timestamp (request) messages received.
sys_IcmpInTimestampReps	The number of ICMP Timestamp Reply messages received.
sys_IcmpInAddrMasks	The number of ICMP Address Mask Request messages received.
sys_IcmpInAddrMaskReps	The number of ICMP Address Mask Reply messages received.
sys_IcmpOutMsgs	The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by IcmpOutErrors.
sys_IcmpOutErrors	The number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers. This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter's value.
sys_IcmpOutDestUnreachs	The number of ICMP Destination Unreachable messages sent.
sys_IcmpOutTimeExcds	The number of ICMP Time Exceeded messages sent.
sys_IcmpOutParmProbs	The number of ICMP Parameter Problem messages sent.
sys_IcmpOutSrcQuenchs	The number of ICMP Source Quench messages sent.
sys_IcmpOutRedirects	The number of ICMP Redirect messages sent.
sys_IcmpOutEchos	The number of ICMP Echo (request) messages sent.
sys_IcmpOutEchoReps	The number of ICMP Echo Reply messages sent.
sys_IcmpOutTimestamps	The number of ICMP Timestamp (request) messages sent.
sys_IcmpOutTimestampReps	The number of ICMP Timestamp Reply messages sent.
sys_IcmpOutAddrMasks	The number of ICMP Address Mask Request messages sent.
sys_IcmpOutAddrMaskReps	The number of ICMP Address Mask Reply messages sent.

Table 37: Predefined CLEAR-Flow Counters (continued)

Counter Name	Description
sys_IcmpInProtoUnreachs	The number of incoming ICMP packets addressed to a not-in-use / unreachable / invalid protocol. This message is in the general category of ICMP destination unreachable error messages.
sys_IcmpInBadLen	The number of incoming bad ICMP length packets.
sys_IcmpInBadCode	The number of incoming ICMP packets with a bad code field value.
sys_IcmpInTooShort	The number of incoming short ICMP packets.
sys_IcmpInBadChksum	The number of incoming ICMP packets with bad checksums.
sys_IcmpInRouterAdv	The number of incoming ICMP router advertisements. Router advertisements are used by IP hosts to discover addresses of neighboring routers.
sys_IcmpOutProtoUnreachs	The number of outgoing ICMP packets addressed to a not-in-use / unreachable / invalid protocol. This message is in the general category of ICMP destination unreachable error messages.
sys_IcmpOutRouterAdv	The number of outgoing ICMP router advertisements. Router advertisements are used by IP hosts to discover addresses of neighboring routers.
sys_IgmpInQueries	The number of Host Membership Query messages that have been received on this interface.
sys_IgmpInReports	The number of Host Membership Report messages that have been received on this interface for this group address.
sys_IgmpInLeaves	The number of incoming *IGMP (Internet Group Management Protocol)* leave requests.
sys_IgmpInErrors	The number of incoming IGMP errors.
sys_IgmpOutQueries	The number of Host Membership Query messages that have been sent on this interface.
sys_IgmpOutReports	The number of Host Membership Report messages that have been sent on this interface for this group address.
sys_IgmpOutLeaves	The number of outgoing IGMP leave requests.

Configuring CLEAR-Flow

CLEAR-Flow is an extension to *ACLs*, so you must be familiar with configuring ACLs before you add CLEAR-Flow rules to your ACL policies. Creating ACLs is described in detail in the ACLs chapter. The chapter describes how to create ACL policies, the syntax of an ACL policy file, and how to apply ACL policies to the switch.

In this chapter, you will find information about the CLEAR-Flow rules that you add to ACL policies, including the CLEAR-Flow rules' syntax and behavior.

[5] The length of an ICMP packet depends on the type and code field.

After creating the ACLs that contain CLEAR-Flow rules, and after applying the ACLs to the appropriate interface, you enable CLEAR-Flow on the switch. When CLEAR-Flow is enabled, the agent on the switch evaluates the rules, and when any rules are triggered, the CLEAR-Flow actions are executed.

- To enable CLEAR-Flow, enter the command:
  ```
  enable clear-flow
  ```
- To disable CLEAR-Flow, enter the command:
  ```
  disable clear-flow
  ```
  When you disable the CLEAR-Flow agent on the switch, CLEAR-Flow sampling stops and all rules are left in the current state.

Note
Any actions triggered while CLEAR-Flow is enabled will continue when CLEAR-Flow is disabled, unless explicitly stopped.

Displaying CLEAR-Flow Configuration and Activity

- Display the state of the CLEAR-Flow agent, any CLEAR-Flow policies on each interface, and the number of CLEAR-Flow rules by entering the command:
  ```
  show clear-flow
  ```
- Display the CLEAR-Flow rules and configuration by entering the command:.
  ```
  show clear-flow rule
  ```
- Display all the rules by entering the command:
  ```
  show clear-flow rule-all
  ```
  When CLEAR-Flow is enabled, any rules that satisfy the threshold will trigger and take action.
- Display the CLEAR-Flow rules that have been triggered by entering the command:.
  ```
  show clear-flow rule-triggered
  ```
- Display which ACLs have been modified by CLEAR-Flow rules.
  ```
  show clear-flow acl-modified
  ```

CLEAR-Flow Rule Examples

In the examples that follow, one to two *ACL* rule entries are followed by a CLEAR-Flow rule entry. The examples illustrate the four CLEAR-Flow rule expressions: count, delta, ratio, and delta-ratio.

Count Expression Example

In the following example, every ten seconds the CLEAR-Flow agent will request the counter1 statistics from the hardware.

After it receives the counter value, it will evaluate the CLEAR-Flow rule. If the value of counter1 is greater than 1,000,000 packets, the CLEAR-Flow agent will send a trap message to the *SNMP* master, and change the *ACL* acl_rule1 to block traffic (acl_rule1 is modified to a deny rule).

Since there is no period configured for the snmptrap statement, the message is sent only once.

```
entry acl_rule1 {
if {
destination-address 192.168.16.0/24;
destination-port 2049;
protocol tcp;
} then {
count counter1;
}
}
entry cflow_count_rule_example  {
if { count counter1 > 1000000 ;
period 10 ;
}
Then {
snmptrap 123 "Traffic on acl_rule1 exceeds threshold";
deny acl_rule1;
}
}
```

Delta Expression Example

In this example, every ten seconds the CLEAR-Flow agent will request the counter1 statistics from the hardware.

After it receives the counter value, it will then evaluate the rule. If the delta (change) of the counter1 value from the last sampled value ten seconds ago is greater than or equal to 1,000 packets, the CLEAR-Flow agent will send a trap message to the *SNMP* master and change the *ACL* acl_rule1 to move the traffic to QP3. In addition, reduce the peak rate to 5 Kbps on QP3. As long as the delta continues to be greater than or equal to 1000 packets, the CLEAR-Flow agent will repeatedly send a trap message every 120 seconds. When the delta falls below the threshold, the agent will execute the two actions in the else portion; it will send a single SNMP trap message, return the traffic to QP1, and reset QP3 to its original bandwidth.

```
entry acl_rule1 {
if {
destination-address 192.168.16.0/24;
destination-port 2049;
protocol tcp;
} then {
count counter1;
}
}
entry cflow_delta_rule_example  {
if  { delta counter1 >= 100000 ;
period 10 ;
} then {
snmptrap 123 "Traffic to 192.168.16.0/24 exceed rate limit" 120;
qosprofile acl_rule1 QP3;
cli "configure qosprofile qp3 peak_rate 5 K ports all" ;
} else {
snmptrap 123 "Traffic to 192.168.16.0/24 falls below rate limit";
qosprofile acl_rule1 QP1;
cli "configure qosprofile qp3 maxbw 100 ports all" ;
```

```
}
}
```

Ratio Expression Example

In this example, every two seconds the CLEAR-Flow agent will request the counter1 and counter2 statistics from the hardware.

After it receives the two counter values, it will then check each counter value against its minimum valid threshold, which is 1,000. If both of the counter values is greater than 1,000, it then calculates the ratio of counter1 and counter2. If the ratio is greater than 5, the agent will execute the actions in the then clause, which consists of logging a message to the syslog server. Before logging the syslog string, the agent will replace the $ruleName keyword with the string cflow_ratio_rule_example, the $ruleValue keyword with the calculated ratio value, and the $ruleThreshold keyword with a value of 5. If either of the counter values is below the minimum value of 1,000, or the ratio is below the threshold of 5, the expression is false and no action is taken.

```
entry acl_rule1 {
if {
protocol udp;
} then {
count counter1;
}
}
entry acl_rule2 {
if {
protocol tcp;
} then {
count counter2;
}
}
entry cflow_ratio_rule_example  {
if  { ratio counter1 counter2 > 5 ;
period 2;
min-value 1000;
}
Then {
syslog "Rule $ruleName threshold ratio $ruleValue exceeds limit $ruleThreshold";
}
}
```

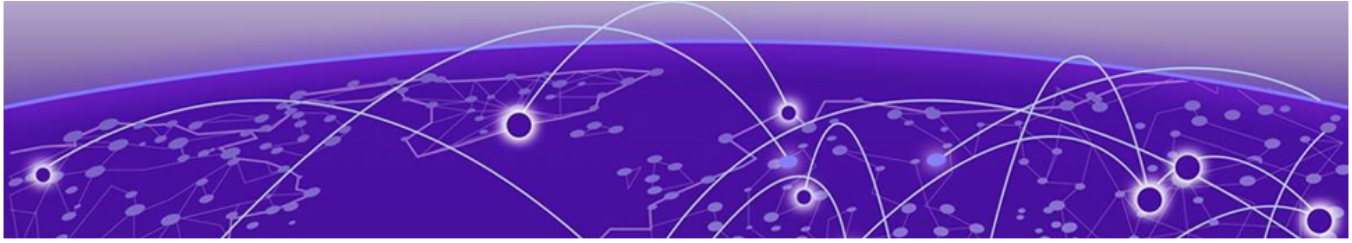
Delta-Ratio Expression Example

In this example, every two seconds, the CLEAR-Flow agent will request the tcpSynCounter and tcpCounter values from the hardware.

After it receives the two counter values, it will first calculate the delta for each of the counters and then check each counter's delta value for its minimum value, which is 100. If both of the counters' delta values are greater than 100, it then calculates the ratio of the delta of two counters. If the ratio is greater than 10, then the agent will log a warning message and deny all SYN traffic on the interface. No period value for the syslog message is given, so the message will be logged once when the expression first becomes true. When the expression transitions from true to false, a different message will be logged and the SYN traffic on the interface will be permitted again. The delta-

ratio value has to fall below a threshold of 8 for the expression to be evaluated to be false.

```
entry acl_syn {
if {
protocol tcp_flags SYN;
} then {
count tcpSynCounter;
}
}
entry acl_tcp {
if {
protocol tcp;
} then {
count tcpCounter;
}
}
entry cflow_delta_ratio_rule_example  {
if  { delta-ratio tcpSynCounter tcpCounter > 10 ;
period 2;
min-value 100;
              threshold 8;
            } then {
syslog "Syn attack on port $port is detected" WARN;
deny acl_syn;
} else {
syslog "Syn attack on port $port is no longer detected" WARN;
permit acl_syn;
}
}
```



Data Center Solutions

Data Center Overview on page 520
Managing the DCBX Feature on page 526
Managing the XNV Feature, VM Tracking on page 527
Managing Direct Attach to Support VEPA on page 547

This chapter provides information about Extreme Network's Data Center Solutions. It provides an overview of data centers and provides information about how to configure and manage data center features, including DCBX, *Extreme Network Virtualization (XNV)*, VM Tracking, and Direct Attach to support VEPA.

Data Center Overview

Typical data centers support multiple Virtual Machines (VMs) on a single server. These VMs usually require network connectivity to provide their services to network users and to other VMs. The following sections introduce ExtremeXOS software features that support VM network connectivity:

Note
For additional information on using ExtremeXOS features to implement Data Center Bridging, see the application note titled *Enhanced Transmission Selection (ETS) Deployment and Configuration for ExtremeXOS* on the Extreme Networks website.

Introduction to Data Center Bridging

DCB (Data Center Bridging) is a set of IEEE 802.1Q extensions to standard Ethernet, that provide an operational framework for unifying Local Area Networks (LAN), Storage Area Networks (SAN) and Inter-Process Communication (IPC) traffic between switches and endpoints onto a single transport layer.

Data Center Bridging Exchange Protocol

The Data Center Bridging eXchange (DCBX) protocol is used by *DCB* devices to exchange DCB configuration information with directly connected peers. In an ExtremeXOS enabled switch, the switch uses DCBX to advertise its DCB configuration to end stations. The end stations can then configure themselves to use the switch DCB services. If the peers do not support a common configuration for one or more features, the switch generates messages to alert network management. The switch does not accept configuration change requests from end stations.

The DCBX protocol advertises the following types of information:

- DCBX version information, so that the peers can negotiate a common version to use.
- Enhanced Transmission Selection (ETS) information for *QoS (Quality of Service)* parameters such as bandwidth allocation per traffic class (802.1p COS), priority for each traffic class, and the algorithm used for servicing traffic classes.
- Priority-based Flow Control (PFC) information for managing flow control between peers.
- Application priority information for prioritizing traffic for special applications that do not map directly to an established traffic class.

The ExtremeXOS software supports two versions of DCBX standards. The first version is a pre-standard version known as the baseline version, or more specifically as Baseline Version 1.01. The DCBX baseline version is specified in DCB Capability Exchange Protocol Base Specification Rev 1.01 and was developed outside of the IEEE and later submitted to the IEEE for standardization. The IEEE agreed to standardize DCBX as part of IEEE 802.1Qaz Enhanced Transmission Selection for Bandwidth Sharing Between Traffic Classes. While IEEE 802.1Qaz has progressed through the standards process, many companies have released support for the baseline version.

After you enable DCBX, the protocol collects most of the information to be advertised from other switch services such as QoS and PFC. The only DCBX feature that needs configuration is the application priority feature.

DCBX uses the *LLDP (Link Layer Discovery Protocol)* (IEEE 802.1AB) to exchange attributes between two link peers. DCBX attributes are packaged into organizationally specific TLVs, which are different for the Baseline and IEEE 802.1Qaz versions. Information on the TLV support differences is provided in the ExtremeXOS Command Reference under the command description for the command: `show lldp {port [all | port_list]} dcbx {ieee|baseline} {detailed}`

Custom Application Support

The DCBX custom application support feature allows you to prioritize and manage traffic flow through the switch based on the application type. This feature allows you to configure DCBX handling of the following applications:

- Fiber Channel Over Ethernet (FCoE)
- Internet Small Computer System Interface (iSCSI)
- Any application that can be defined by:
  - Ethertype value
  - Layer 4 port number
  - TCP port number
  - UDP port number

When you configure a custom application, you define a priority number that applies to traffic related to that application. DCBX advertises this priority to end stations in an application TLV. End stations that support this feature use the priority number for communications with the switch. The priority number maps to an 802.1p value, which determines which *QoS* profile in the switch manages the application traffic.

The software supports a maximum of eight application configurations.

Priority-based Flow Control

Priority flow control (PFC) is defined in the IEEE 802.1Qbb standard as an extension of the IEEE 802.3x flow control standard. When buffer congestion is detected, IEEE 802.3x flow control allows the communicating device to pause all traffic on the port, whereas PFC allows the device to pause just a portion of the traffic and allow other traffic on the same port to continue.

The rest of this section provides general guidelines for configuring the ExtremeXOS PFC feature for *DCB* operation. After you configure PFC, DCBX advertises the PFC compatible configuration to DCBX peers on all DCBX enabled ports.

- PFC configuration is controlled per-port using the following command:

  `enable flow-control [tx-pause {priority priority} | rx-pause {qosprofile qosprofile}] ports [all | port_list]`

  The rx-pause option is configured on the *QoS* profile.

  The PFC priority to which a QoS profile responds is fixed and is determined by the QoS profile number such that qpN responds to a PFC frame for priority N-1.

  For example, the following command enables PFC priority 4 for qp5 on ports 1-24:

  ```
  enable flow-control rx-pause qosprofile qp5 ports 1-24
  ```

  After the above command is entered, if a PFC frame is received indicating that priority 4 should be paused, then qp5 will be paused. Note that qp5 is paused regardless of whether the packets mapped to qp5 have priority 4 or other priorities. For example, if we enter the command configure dot1p type 3 qosprofile qp5, priority 3 packets are queued in qp5, and a PFC pause frame for priority 4 pauses priority 3 frames, which might not be desired. For this reason, you should be careful about mapping multiple priorities to the same QoS profile when PFC is enabled for that profile.

The tx-pause option is configured on the priority itself. For example, the following command enables the transmittal of PFC Pause frames for priority 4 when frames with priority 4 are congested:

```
enable flow-control tx-pause priority 4 ports 1-24
```

The tx-pause configuration determines what is advertised in the DCBX PFC TLV. In order for PFC to work correctly, it is important to ensure that all switches in the DCB network are receiving and transmitting PFC consistently for each priority on all ports.

In summary, the following three commands ensure that PFC is enabled for priority 4 traffic on ports 1-24:

```
configure dot1p type 4 qosprofile qp5
enable flow-control rx-pause qosprofile qp5 ports 1-24
enable flow-control tx-pause priority 4 ports 1-24
```

For more information on PFC, see IEEE 802.1Qbb Priority Flow Control on page 234.

Introduction to the XNV Feature

The *XNV* feature, which is also known as Virtual Machine (VM) tracking, enables the ExtremeXOS software to support VM port movement, port configuration, and inventory on network switches. VM movement and operation on one or more VM servers is managed by a VM Manager (VMM) application. The XNV feature enables a network switch to respond to VM movement and report VM activity to network management software.

VM network access support enables a switch to support VMs as follows:

- Identify a VM by its MAC address and authenticate the VM connection to the network.
- Apply a custom port configuration in response to VM authentication.
- Remove a custom port configuration when a VM *FDB (forwarding database)* entry ages out.
- Detect a VM move between switch ports or switches and configure the old and new ports appropriately.

To support VM mobility, the XNV feature requires that each VM use unique, static MAC and IP addresses. Switch port operation for a VM can be configured with a policy file or an *ACL (Access Control List)*.

VM Port Configuration

An important part of the *XNV* feature is the ability to configure a switch port to support a particular VM. A Virtual Port Profile (VPP) identifies a policy file or *ACL* rule to associate with a VM entry in the authentication database. You can define both ingress and egress policies in VPPs to configure a port separately for each direction. When the VPP is configured for a VM entry and the VM is detected on a port, any associated policy or rule is applied to the port in the specified direction.

The XNV feature supports two types of VPPs, Network VPPs (NVPPs) and Local VPPs (LVPPs).

NVPPs are stored on an FTP server called a repository server. The XNV feature supports file synchronization between XNV-enabled switches and the repository server. One of the advantages of the repository server is centralized storage for NVPPs.

LVPPs must be configured on each switch. LVPPs are a good choice for simple network topologies, but NVPPs offer easier network management for more complex network topologies.

VM Authentication Process

The *XNV* feature supports three methods of authentication:

- NMS server authentication.
- Network authentication using a downloaded authentication database stored in the VMMAP file.
- Local authentication using a local database created with ExtremeXOS CLI commands.

The default VM authentication configuration uses all three methods in the following sequence: NMS server (first choice), network based VMMAP file (second choice), and finally, local database. If a service is not available, the switch tries the next authentication service in the sequence.

NMS Server Authentication

If NMS server authentication is enabled and a VM MAC address is detected on a VM-tracking enabled port, the software sends an Access-Request to the configured NMS server for authentication. When the switch receives a response, the switch does one of the following:

- When an Access-Accept packet is received with an NVPP specified, the policies are applied on VM enabled port.
- When an Access-Accept packet is received and no NVPP is specified, the port is authenticated and no policy is applied to the port.
- When an Access-Reject packet is received, the port is unauthenticated and no policy is applied.
- When an Access-Reject packet indicates that the NMS server timed-out or is not reachable, the switch tries to authenticate the VM MAC address based on the next authentication method configured, which can be either network authentication or local authentication.

The Access-Accept packet from the NMS server can include the following Vendor Specific Attributes (VSAs):

- VM name
- VM IP address
- VPP configured for the VM

An Access-Reject packet contains no VSA.

Network (VMMAP) Authentication

If network (VMMAP) authentication is enabled and a VM MAC address is detected on a VM-tracking enabled port, the switch uses the VMMAP file to authenticate the VM and applies the appropriate VPP.

Local Authentication

If local authentication is enabled and a VM MAC address is detected on a VM-tracking enabled port, the switch uses the local database to authenticate the VM and apply the appropriate VPP.

Authentication Failure

If all configured authentication methods fail, EMS messages are logged and no VPP is applied.

Possible remedies include:

- Fix the authentication process that failed. Look for misconfiguration or down segments.
- Configure UPM to take action on the related EMS message.

- If one or two authentication methods are configured, configure additional authentication methods.

Duplicate VM MAC Detected

Each VM MAC must be unique. If duplicate MAC addresses are detected on the switch, whether on the same *VLAN (Virtual LAN)* or different VLANs, the switch supports only the last MAC detected.

File Synchronization

The *XNV* feature supports file synchronization between XNV-enabled switches and the repository server. The files stored on the repository server include the .map, .vpp, and .pol files. One of the advantages of the repository server is that multiple XNV-enabled switches can use the repository server to collect the network VM configuration files. The XNV feature provides for access to a secondary repository server if the primary repository server is unavailable.

Through file synchronization, the network files are periodically downloaded to the XNV-enabled switches, which allows these switches to continue to support VM authentication when the NMS server is unavailable.

Network Management and Inventory

The *XNV* feature is designed to support network management programs. The ExtremeXOS software contains *SNMP (Simple Network Management Protocol)* MIBs, which allow network management programs to view VM network configuration data, discover the VM inventory, and make configuration changes.

For instructions on managing the XNV feature using the switch CLI, see Managing the XNV Feature, VM Tracking on page 527.

Introduction to the Direct Attach Feature

The direct attach feature is a port configuration feature that supports VM-to-VM communication on a directly connected server that uses the Virtual Ethernet Port Aggregator (VEPA) feature on that server. Without VEPA and direct attach, a VM server must use a virtual Ethernet bridge or switch on the VM server to enable Ethernet communications between VMs. With VEPA, the VM server can rely on a directly connected switch to receive and reflect VM-to-VM messages between VMs on the same server.

The ExtremeXOS direct attach feature works with VEPA software on a VM server to intelligently forward unicast, flood, and broadcast traffic. Without direct attach, frames are never forwarded back out the same port on which they arrive. With direct attach, frames can be forwarded back out the ingress port, and VEPA software on the VM server ensures that the frames are forwarded appropriately.

For instructions on managing the Direct Attach feature, see Managing Direct Attach to Support VEPA on page 547.

Managing the DCBX Feature

Enabling DCBX on Switch Ports

DCBX uses *LLDP* to advertise *DCB* capabilities to DCB peers.

Use the following commands to enable LLDP and the DCBX feature on switch ports:

```
enable lldp ports [all | port_list] {receive-only | transmit-only}

configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific dcbx {ieee|baseline}
```

Configuring DCBX Application Priority Instances

Each DCBX application priority instance maps traffic from one of the supported application types to a TC or PG priority, which selects a specific *QoS* profile for traffic management. Supported application types include:

- Fiber Channel Over Ethernet (FCoE)
- Internet Small Computer System Interface (iSCSI)

Use the following commands to add or delete DCBX application priority instances:

```
configure lldp ports [all | port_list] dcbx add application [name
application_name | ethertype ethertype_value | L4-port port_number |
tcp-port port_number | udp-port port_number] priority priority_value

configure lldp ports [all | port_list] dcbx delete application [all-
applications | name application_name | ethertype ethertype_value | L4-
port port_number | tcp-port port_number | udp-port port_number]
```

Displaying DCBX Configuration and Statistics

Use the following commands to display DCBX feature configuration and statistics:

```
show lldp {port [all | port_list]} {detailed}

show lldp {port [all | port_list]} dcbx {ieee|baseline} {detailed}
```

DCBX Configuration Example

The following is a sample DCBX configuration:

```
enable lldp ports 1
configure lldp port 1 advertise vendor-specific dcbx ieee
configure lldp port 1 advertise vendor-specific dcbx baseline

enable lldp ports 2
configure lldp port 2 advertise vendor-specific dcbx ieee
configure lldp port 2 advertise vendor-specific dcbx baseline
```

```
configure lldp ports 1 dcbx add application name iscsi priority 4
configure lldp ports 1 dcbx add application name fcoe priority 3
configure lldp ports 2 dcbx add application name iscsi priority 4
configure lldp ports 2 dcbx add application name fcoe priority 3
configure lldp ports 2 dcbx add application L4-port 25 priority 4
configure lldp ports 2 dcbx add application tcp-port 4500 priority 4
configure lldp ports 2 dcbx add application udp-port 45 priority 5
configure lldp ports 2 dcbx add application ethertype 2536 priority 4
```

Managing the XNV Feature, VM Tracking

Limitations

The following features of Change-of-Authorization (RFC5176) are not implemented in ExtremeXOS:

- Reverse Path Forwarding Check—Typically this is used in a proxy scenario. This check is used to determine if the IP address indicated by the *RADIUS (Remote Authentication Dial In User Service)* attributes is a routable destination address for a request sent by the switch software.
- IPSEC encryption—End-to-end encryption of both the RADIUS requests and responses.
- Disconnect-Request and Change-of-Authorization packets identifying sessions with anything other than the Calling-Station-Id attribute containing a properly formatted MAC address. In addition to the Calling-Station-ID attribute, you can also use a NAS-Port attribute, which indicates the index of the specific port the session is connected to.
- Acct-Session-Id attribute—This is an alternate means of session identification. Sessions are currently uniquely identified by port and MAC address pair.
- Retransmissions of Disconnect-Request or Change-of-Authorization ACK and NAK packets—Retransmissions of packets is the responsibility of the device initiating the dynamic authorization transactions.

Managing VM Tracking on the Switch

Use the following steps to manage VM tracking on the switch:

- Issue the following command to enable the VM tracking feature on the switch:

  ```
  enable vm-tracking
  ```

- Issue the following command to disable the VM tracking feature on the switch:

  ```
  disable vm-tracking
  ```

  Note
  When the VM tracking feature is disabled, file synchronization with the repository server stops.

- Issue the following command to view the VM tracking feature configuration and the list of authenticated VMs:

  show vm-tracking

Managing VM Tracking on Specific Ports

Before you enable the VM tracking feature on specific ports, you must enable VM tracking on the switch, configure the authentication method and sequence, and the VM authentication databases.

- When this configuration is complete, you can use the following command to enable VM tracking on one or more ports:

  `enable vm-tracking ports port_list`

- To disable the VM tracking feature on a group of ports, use the following command:

  `disable vm-tracking ports port_list`

- To view the VM tracking feature configuration on one or more ports, use the following command:

  `show vm-tracking port port_list`

Configuring the Authentication Method and Sequence

You can configure VM authentication through the following services:

- NMS server
- Network based VMMAP file
- Local database

The default VM authentication configuration uses all three methods in the following sequence: NMS server (first choice), network based VMMAP file (second choice), and finally, local database. If a service is not available, the switch tries the next authentication service in the sequence.

To configure one or more authentication methods and a preferred sequence, use the following command:

```
configure vm-tracking authentication database-order [[nms] | [vm-map] |
[local] | [nms local] | [local nms] | [nms vm-map] | [vm-maplocal] |
[local vm-map] | [nms vm-map local] | [localnmsvm-map]]
```

XNV and MLAG

Starting with ExtremeXOS 15.7, as long as peers have ISC connectivity, only one of the MLAG peers authenticates a VM that is learned on an MLAG port.

- When ISC connectivity between the MLAG peers is established, the peer with the highest IP address is chosen to be the authenticator. This peer will authenticate a VM based on the chosen authentication method.
- The result of the authentication is checkpointed by the authenticator to its peer so that the same VPP gets applied to the VM on both peers.
- When the MLAG peer that is the authenticator goes down, the other peer detects that the authenticator is down and re-authenticates the VM at the next authentication interval. Note that the peer that takes over as the authenticator does not re-authenticate the VMs immediately but waits for the re-authentication timer to expire.

- VMs learned on non-MLAG ports are authenticated by the detecting peer.
- All authentication-related configurations like *RADIUS* address, repository for VMMAP, local DB, etc. must be identical on both peers. This is an existing requirement and there is no change to this requirement.

XNV Dynamic VLAN

Starting in release 15.3, when a virtual machine is detected, ExtremeXOS dynamically creates the *VLAN* that is required for the VM to send traffic. If a virtual machine shuts down or is moved, its VLAN is pruned to preserve bandwidth. This feature creates an adaptive infrastructure in which the network responds to changes dynamically in the virtual machine network.

Enabling/Disabling XNV Dynamic VLAN

Enabling the *XNV* dynamic VLAN feature must be done on a per-port basis. XNV requires that the port on which dynamic VLANs is enabled is part of the "default or "base" VLAN as untagged. This "default", or "base", VLAN for the port is the VLAN on which untagged packets are classified to when no VLAN configuration is available for the MAC. This default VLAN should be present, and you should manually add the port to this VLAN before you enable the feature. Enabling this feature on a port results in a failure if any of the following conditions are true:

- If XNV is not enabled, the command only results in a warning, and does not fail. XNV can be enabled later.
- The port is not an untagged member of any VLAN.

When a VLAN's MAC is detected on a port, XNV consults the configuration database to determine the VLAN configuration for the VM. For a case where the VM sends tagged traffic, the VLAN tag of the received frame is used to determine VLAN classification for the VM's traffic. If VLAN configuration exists for the VM and it conflicts with the actual tag present in received traffic, XNV reports an EMS message and does not trigger VLAN creation or port addition. However, if no configuration is present for the VM, XNV assumes that there are no restrictions for classifying traffic for the VM to the received VLAN.

For untagged traffic, XNV can determine the VLAN for the VM from any one of the three possible sources:

- VLAN configuration for the VM MAC entry.
- VLAN configuration for the VPP associated with the VM's MAC. The VPP can either be a network VPP or a local VPP.
- In case of untagged traffic from the VM, the "default" VLAN for the port that is specified as part of the dynamic VLAN enable configuration.

This list determines the order of precedence for VLAN classification for untagged traffic only. For tagged VLAN traffic, XNV validates the tag of the received traffic with then VLAN tag configuration for that VM.

In addition to the VLAN tag, you can specify the VR to which the dynamically created VLAN needs to be associated. The VR configuration is relevant only if a VLAN tag is configured for the VM.

Table 38: Associating Dynamically Created VLANS to VRs

Configured VR on Port	Configured VR for VM (from VM Mapping Entry or VPP)	VLAN Already Exists on the Switch	Dynamic VLANs VR
None	None	No	*VR-Default*
None	None	Yes	VLAN's VR
None	VR-X	No	VR-X (Configured VR for VM) if VR-X is valid.) Otherwise an EMS error is displayed indicating the VR-X is invalid.
None	VR-X	Yes	VLAN's VR. An EMS error is displayed if the VLAN's VR is not VR-X.
VR-X	None	No	VR-X (Port's VR).
VR-X	None	Yes	VR-X if VLAN's VR is VR-X. If it is not, an EMS error is displayed indicating the VR-X is invalid.
VR-X	VR-Y	No	Dynamic VLAN is not created when Port Level VR and VM-MAC VR are different, and *FDB* is learned on a system generated VMAN. An EMS warning is generated on the switch log, because a Dynamic VLAN cannot be created.
VR-X	VR-Y	Yes	VR-X if VLAN is part of VR-X. Otherwise, EMS error is displayed.

When you disable dynamic VLAN on a port, XNV does the following:

- Triggers deletion of MAC-based entries on that port in the hardware.
- If the port has been added to any VLAN by XNV, XNV triggers a flush for those VLANs.

- If the port has been added to an VLAN by XNV, XNV requests VLAN manager to remove the port from the VLAN.

  Note
  It is up to the VLAN manager to decide if the port actually needs to be removed from the VLAN.

On deleting the ports from base/default VLAN the below warning message will be thrown and XNV Dynamic vlan gets disabled on that port:

```
 Warning: Removing the untagged VLAN from a port may disrupt network connectivity. IDM and
VMT may not be functional on the port without an untagged VLAN.
```

Note
This behavior is in effect from ExtremeXOS 16.1.

Example

```
create vlan v1
con v1 add ports 1untagged
enable vm-tracking
enable vm-tracking ports 1
enable vm-tracking dynamic-vlan ports 1
con vlan v1 delete ports 1
Warning: Removing the untagged VLAN from a port may disrupt network connectivity. IDM
and VMT may not be functional on the port without an untagged VLAN.

show vm-tracking
-------------------------------------------------------------
VM Tracking Global Configuration
-------------------------------------------------------------
VM Tracking                       : Enabled
VM  Tracking authentication order: nms vm-map local
VM Tracking nms reauth  period    : 0 (Re-authentication disabled)
VM Tracking blackhole policy        : none
-------------------------------------------------------------

Port                          : 1
VM Tracking                   : Enabled
VM Tracking Dynamic VLAN      : Disabled
```

When XNV is disabled on a port, the XNV dynamic VLAN feature is also disabled. The XNV dynamic VLAN configuration is not persistent, and needs to be re-enabled after XNV is re-enabled on that port.

Tracking XNV Per VM Statistics

Beginning in release 15.3, each local and network VPP has the option to specify whether a counter needs to be installed to count traffic matching the virtual machine MAC which gets the VPP mapping. You can choose to install a counter to collect statistics for ingress traffic only, egress traffic only, or traffic in both directions.

Once the ingress counter installation option is selected for a specific local or network VPP and the virtual machine which has this VPP mapping is detected on the switch, the counter is installed with the name "xnv_ing_dyn_cnt_vmxxxxxxxxxxxx" for the port on which the VM MAC is detected. In this case, xxxxxxxxxxxx denotes the virtual

machine MAC for which the counter is installed. In the same way, the egress counter is installed using the name "xnv_egr_dyn_cnt_vmxxxxxxxxxxx" for that port.

You can view a list of packet/byte counts for this counter name using the command `show access dynamic-counter`. The counter is uninstalled only when the virtual machine MAC is deleted on the switch or the VPP is mapped to a virtual machine MAC which has the counter option set to none. If the VM MAC move happens then the counter installed on the previous port is uninstalled and the counter is installed on the new port. The counter values are not maintained during the MAC move.

Managing the Repository Server

Selecting the Repository Server Directory

All files for NMS and network authentication must be placed in the configured repository server directory. These files include the following:

- MANIFEST
- VMMAP
- NVPP policy files

By default, the *XNV* feature tries to access the FTP server with anonymous login and fetch the files from the pub directory within the FTP server root directory.

To configure a different directory for repository server files, use the following command:

```
configure vm-tracking repository
```

Creating the MANIFEST File

The MANIFEST file identifies the VMMAP, NVPP, and policy files that are to be used for either NMS or network authentication. The MANIFEST file is downloaded to the switch at the specified refresh interval. Each time the MANIFEST file is downloaded, the switch scans the file and compares the file entries and timestamps to those files on the switch. If the switch detects newer files, it downloads those files to the switch.

You can create the MANIFEST file with a text editor. The MANIFEST file must be placed on the repository server as described in Selecting the Repository Server Directory on page 532.

The format of MANIFEST files is:

```
File1 yyyy-mm-dd hh:mm:ss
File2 yyyy-mm-dd hh:mm:ss
```

Because the definition for each file in the MANIFEST includes a date and time, you must update the MANIFEST file every time you update the VMMAP file or a policy file.

The following is a sample MANIFEST file:

```
a1.map 2010-07-07 18:57:00
a1.vpp 2010-07-07 18:57:00
```

```
a2.map 2010-07-07 18:57:00
a2.vpp 2010-07-07 18:57:00
policy1.pol 2010-07-07 18:57:00
epolicy1.pol 2010-07-07 18:57:00
```

The file extensions for the files in the MANIFEST file identify the supported file types:

- .map—VMMAP files
- .vpp—VPP files
- .pol—Policy files

Creating a VMMAP File

Use a text editor to create a VMMAP file. VMMAP file entries must use the following XML format:

```
<VMLIST>
    <VM>
        <MAC>00:00:00:00:00:21</MAC>
        <NAME>network_vm1</NAME>
        <IPV4>10.10.10.10</IPV4>
        <VPP>nvpp1</VPP>
    </VM>
    <VM>
        <MAC>00:00:00:00:00:22</MAC>
        <NAME>network_vm2</NAME>
        <IPV4>20.20.20.20</IPV4>
        <VPP>nvpp2</VPP>
    </VM>
</VMLIST>
```

When creating VMMAP file entries, use the following guidelines:

- The VPP file supports up to 400 child nodes.
- The MAC address is required.
- If you do not want to specify a VM name, specify none.
- If you do not want to specify an IP address, specify 0.0.0.0.
- If you do not want to specify a VPP name, specify none.
- If a value such as the VM name contains any space characters, the entire value must be specified between double quotation marks (").

For information on where to place the VMMAP file, see Selecting the Repository Server Directory on page 532.

Creating VPP Files

Use a text editor to create a VPP file. VPP file entries must use the following XML format:

```
<vppList>
  <vpp>
    <name>nvpp1</name>
      <last-updated>2002-05-30T09:00:00</last-updated>
    <policy>
      <name>policy1</name>
      <direction>ingress</direction>
      <order>1</order>
```

```
    </policy>
    <policy>
      <name>policy4</name>
      <direction>ingress</direction>
      <order>4</order>
    </policy>
    <policy>
      <name>epolicy1</name>
      <direction>egress</direction>
      <order>1</order>
    </policy>
    <policy>
      <name>epolicy4</name>
      <direction>egress</direction>
      <order>4</order>
    </policy>
  </vpp>
</vppList>
```

The VPP file supports up to 400 child nodes, and each VPP entry supports up to eight ingress and four egress *ACL* or policies. If multiple policies are defined within a VPP entry for either ingress or egress, the switch uses the policy with the lowest order number. If two ingress or egress policies have the same order number, the switch selects the policy based on which name is lexicographically lower.

To refresh all policies which are all associated and applied to each VPP, use the following command:

```
refresh policy policy-name
```

The NVPP policy files must be placed on the repository server as described in Selecting the Repository Server Directory on page 532.

Creating Policy Files

For instructions on creating policy files, see Policy Manager on page 1448.

To display the policy file or *ACL* associated with one or all VPPs, use the following command:

```
show vm-tracking vpp {vpp_name}
```

Managing Switch Access to the Repository Server

- To enable and configure file synchronization between an *XNV*-enabled switch and a repository server, use the following command:

  ```
  configure vm-tracking repository [primary | secondary] server
  [ipaddress | hostname] {vr vr_name} {refresh-interval seconds} {path-
  name path_name} {user user_name {encrypted} password}
  ```

- To force file synchronization with the repository server, use the following command:

  ```
  run vm-tracking repository sync-now
  ```

- To remove the configuration for one or both repository servers, use the following command:

  ```
  unconfigure vm-tracking repository {primary | secondary}
  ```

- To display the repository server configuration and status, use the following command:

  ```
  show vm-tracking repository {primary | secondary}
  ```

Manage NMS Server Authentication

NMS server authentication uses the *RADIUS* protocol to authenticate VM access to the network with the RADIUS server.

To use NMS authentication, you must do the following:

- Select NMS authentication as described in Configuring the Authentication Method and Sequence on page 528.
- Prepare the network repository server as described in Managing the Repository Server on page 532.
- Configure the NMS client software in the switch as described in Configure the NMS Client Software on page 536.
- Configure the NMS server as described in Configuring the NMS Server Software on page 535.

You can display NMS authenticated VMs as described in Displaying NMS Authenticated VMs on page 536.

Configuring the NMS Server Software

The *RADIUS* server can be used for NMS server authentication. To configure this server, do the following:

1. Add the IP address of each *XNV*-enabled switch as a RADIUS client.
2. Add each VM MAC address as a username (in upper case and should not contain semicolon) and add the MAC address as the password.
3. Add a remote access policy with the Extreme Networks VSAs:
   - Vendor code: 1916
   - VSA ID: 213 (EXTREME_VM_NAME)

     Example: MyVM1
   - VSA ID: 214 (EXTREME_VM_VPP_NAME)

     Example: nvpp1
   - VSA ID: 215 (EXTREME_VM_IP_ADDR)

     Example: 11.1.1.254

Configure the NMS Client Software

- The switch uses NMS client software to connect to an NMS server for VM authentication. Use the following commands to configure the NMS client software in the switch:

  ```
  configure vm-tracking nms [primary | secondary] server [ipaddress | hostname] {udp_port} client-ip client_ip shared-secret {encrypted} secret {vr vr_name}
  ```

  ```
  configure vm-tracking nms timeout seconds
  ```

  ```
  configure vm-tracking timers reauth-period reauth_period
  ```

- To remove the NMs client configuration for one or both NMS servers, use the following command:

  ```
  unconfigure vm-tracking nms {server [primary | secondary]}
  ```

- To display the NMS client configuration, use the following command:

  ```
  show vm-tracking nms server {primary | secondary}
  ```

Displaying NMS Authenticated VMs

To display the VMs and corresponding policies in the NMS authentication database, use the following command:

```
show vm-tracking network-vm
```

Managing Network Authentication (Using the VMMAP File)

To use network authentication, you must do the following:

1. Select network authentication as described in Configuring the Authentication Method and Sequence on page 528.
2. Prepare the network repository server as described in Managing the Repository Server on page 532.

To display the VMs and corresponding policies in the network authentication database, use the following command:

```
show vm-tracking network-vm
```

Manage Local Database Authentication

To use local database authentication, you must do the following:

1. Select local database authentication as described in Configuring the Authentication Method and Sequence on page 528.
2. Create and manage local VPPs (LVPPs) as described in Managing the Local VPP Database on page 537.
3. Create VM entries as described in Managing VM Entries in the Local Authentication Database on page 537.

Managing the Local VPP Database

Only one dynamic *ACL* or policy can be added to a VPP. Ingress LVPPs apply to traffic flowing from the VM, into the switch port, and then to the client. Egress LVPPs apply to traffic flowing from the client, out the switch port, and to the VM.

For instructions on creating policy files, see Policy Manager on page 1448. For instructions on creating dynamic ACLs, see ACLs on page 341.

- To create and configure entries in the LVPP database, use the following commands:

  ```
  create vm-tracking vpp vpp_name
  configure vm-tracking vpp vpp_name add [ingress | egress] [policy policy_name | dynamic-rule rule_name] {policy-order policy_order}
  ```

- To delete or unconfigure entries in the local VPP database, use the following commands:

  ```
  delete vm-tracking vpp {vpp_name}
  unconfigure vm-tracking vpp vpp_name
  ```

- To display the policy file or ACL associated with one or more VPPs , use the following command:

  ```
  show vm-tracking vpp {vpp_name}
  ```

Managing VM Entries in the Local Authentication Database

- To create and configure entries in the local authentication database, use the following commands:

  ```
  create vm-tracking local-vm mac-address mac {name name | ipaddress ipaddress vpp vpp_name }
  configure vm-tracking local-vm mac-address mac [name name | ip-address ipaddress | vpp vpp_name]
  ```

- To remove a configuration parameter for a local authentication database entry, or to remove an entry, use the following commands:

  ```
  unconfigure vm-tracking local-vm mac-address mac [name | ip-address | vpp]
  delete vm-tracking local-vm {mac-address mac}
  ```

- To display the local VPP database entries, use the following command:

  ```
  show vm-tracking local-vm {mac-address mac}
  ```

Example XNV Configuration

Figure 38 displays a sample *XNV* topology that will be used for the examples in the following sections:

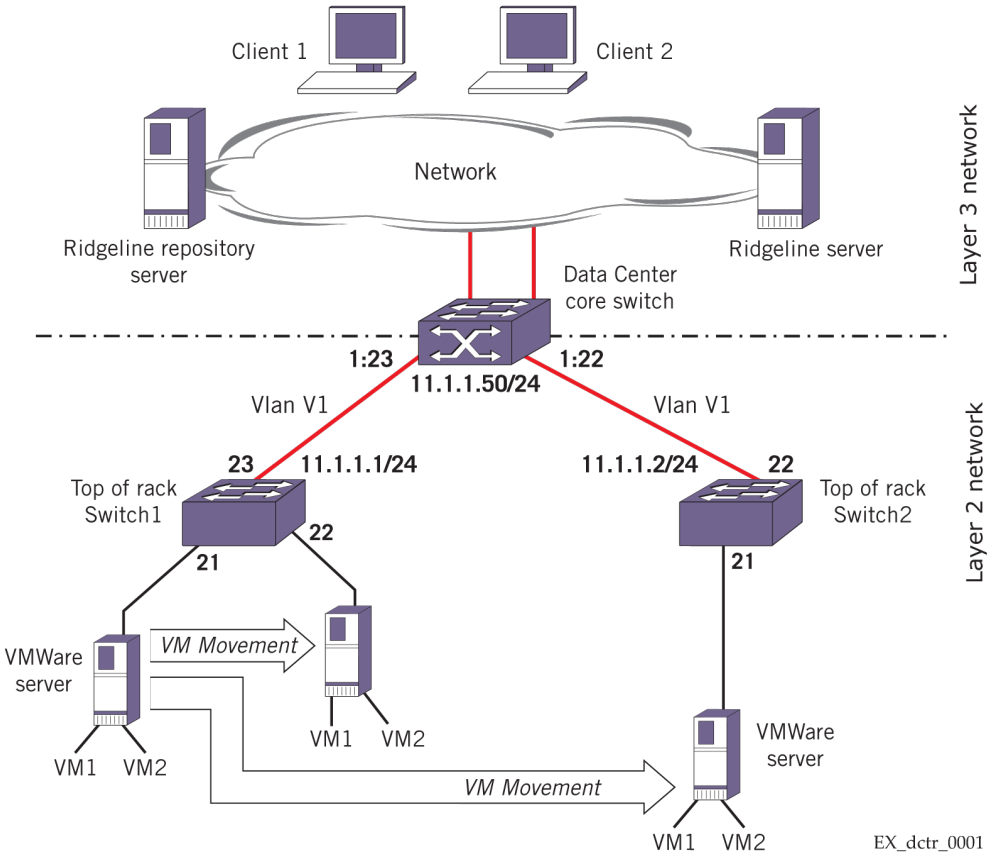


Figure 38: Sample XNV Topology

The example configuration supports the following:

- VM authentication using NMS server, network, or local authentication
- Ingress and egress port configuration for each VM
- VM movement from one switch port to another
- VM movement from one switch to another

Note
Ingress ACLs or policies apply to traffic flowing from the VM, into the switch port, and then to the client. Egress ACLs apply to traffic flowing from the client, out the switch port, and to the VM.

MAC and IP Addresses

The following are the MAC and IP addresses for the example topology:

```
VM1 MAC address: 00:04:96:27:C8:23
VM2 MAC address: 00:04:96:27:C8:24
VM1 IP address: 11.1.1.101
VM2 IP address: 11.1.1.102
Client1 MAC address: 00:04:96:00:00:01
Client2 MAC address: 00:04:97:00:00:02
Repository server IP address: 10.127.8.1
NMS server IP address: 10.127.5.221
```

General VLAN Configuration

The following is the core switch *VLAN* configuration:

```
create vlan v1
configure vlan v1 tag 100
configure vlan v1 add ports 1:22,1:23 tagged
configure vlan v1 ipaddress 11.1.1.50/24
```

The following is the Switch1 VLAN configuration:

```
create vlan v1
configure vlan v1 tag 100
configure vlan v1 add ports 21,22, 23 tagged
configure vlan v1 ipaddress 11.1.1.1/24
```

The following is the Switch2 VLAN configuration:

```
create vlan v1
configure vlan v1 tag 100
configure vlan v1 add ports 21,22 tagged
configure vlan v1 ipaddress 11.1.1.2/24
```

Note
For NMS server and network authentication, the NMS server and repository server must be accessible to all *XNV*-enabled switches through *VR-Mgmt*.

VMWare Server Setup

The VMWare servers must be connected to Switch1 and Switch2 and should have dual quad-core processors. The VMWare servers require the following software:

- VMWare server: ESXi license
- Vsphere EXSI client
- V-Center client

Each physical VMWare server should be configured with two VMs. Use the V-Center client to trigger Vmotion.

Repository Server Setup

The repository server setup for this topology is the same for NMS server authentication and network authentication. The following shows the FTP server setup:

```
FTP login: anonymous
Password: "" (no password)
Repository directory path: pub
[root@linux pub]# pwd
/var/ftp/pub
```

The following is an example MANIFEST file:

```
vm.map 2011-05-11 18:57:00
vpp.vpp 2011-05-11 18:57:00
nvpp1.pol 2011-05-11 18:57:00
nevpp1.pol 2011-05-11 18:57:00
nvpp2.pol 2011-05-11 18:57:00
nevpp2.pol 2011-05-11 18:57:00
```

The following is an example VMMAP file named vm.map:

```
<VMLIST>
  <VM>
    <MAC>00:04:96:27:C8:23</MAC>
    <NAME>vm_1</NAME>
    <IPV4>11.1.1.101</IPV4>
    <VPP>nvpp1</VPP>
    <CTag>1000</CTag>
    <VRName>Vr-Default</VRName>
  </VM>
  <VM>
    <MAC>00:04:96:27:C8:24</MAC>
    <NAME>vm_2</NAME>
    <IPV4>11.1.1.102</IPV4>
    <VPP>nvpp2</VPP>
  </VM>
</VMLIST>
```

The following is an example VPP file named vpp.vpp:

```
<vppList>
  <vpp>
    <name>nvpp1</name>
      <last-updated>2011-05-30T09:00:00</last-updated>
    <policy>
      <name>nvpp1.pol</name>
      <direction>ingress</direction>
      <order>1</order>
    </policy>
    <policy>
      <name>nevpp1.pol</name>
      <direction>egress</direction>
      <order>1</order>
      <CTag>1000</CTag>
      <VRName>Vr-Default</VRName>
    </policy>
  </vpp>
  <vpp>
    <name>nvpp2</name>
      <last-updated>2011-05-30T09:00:00</last-updated>
    <policy>
      <name>nvpp2.pol</name>
      <direction>ingress</direction>
      <order>1</order>
      </policy>
    <policy>
      <name>nevpp2.pol</name>
      <direction>egress</direction>
      <order>1</order>
    </policy>
  </vpp>
</vppList>
```

The following is the nvpp1.pol file:

```
entry nvpp1 {
if match all {
    ethernet-destination-address 00:04:96:00:00:00 / ff:ff:ff:00:00:00 ;
} then {
  deny ;
```

```
  count host1
} }
```

The following is the nvpp2.pol file:

```
entry nvpp2 {
if match all {
    ethernet-destination-address 00:04:97:00:00:00 / ff:ff:ff:00:00:00 ;
} then {
  deny ;
  count host2
} }
```

The following is the nevpp1.pol file:

```
entry nevpp1 {
if match all {
    ethernet-source-address 00:04:96:00:00:00 / ff:ff:ff:00:00:00 ;
} then {
  deny ;
  count h1
} }
```

The following is the nevpp2.pol file:

```
entry nevpp2 {
if match all {
  ethernet-source-address 00:04:97:00:00:00 / ff:ff:ff:00:00:00 ;
} then {
  deny ;
  count h2
} }
```

Example ACL Rules

The following are some example *ACL* rules:

```
entry etherType1 {
    if {
      ethernet-source-address 00:a1:f1:00:00:01;
      }
    then {
      permit;
      count etherType1;
      }
    }
entry denyall {
    if {
        source-address 10.21.1.1/32;
       }
    then {
          deny;
         }
    }
entry allowall {
    if {
        source-address 11.1.1.1/32;
        source-address 12.1.0.0/16;
       }
    then {
```

```
        allow;
       }
    }
entry destIp {
    if {
        destination-address 192.20.1.0/24;
        protocol UDP;
       }
    then {
        deny;
      count destIp;
       }
    }
entry denyAll {
    if {
       }
    then {
          deny;
          count denyAll;
         }
    }
```

General Switch XNV Feature Configuration

The following configuration enables the *XNV* feature on the switch and the specified ports:

```
enable vm-tracking
enable vm-tracking ports 21-22
```

Local VM Authentication Configuration

If you only want to use local authentication, configure the *XNV*-enabled switches as follows:

```
configure vm-tracking authentication database-order local
```

To enable dynamic *VLAN*, issue the following command:

```
enable vm-tracking dynamic-vlan ports 19
```

To add Uplinkports to Dynamic VLAN:

```
configure vlan dynamic-vlan uplink-ports add ports port_no
```

To delete the uplink port:

```
configure vlan dynamic-vlan uplink-ports delete ports port_no
```

The following is the policy1.pol file for Port 21 in the ingress direction:

```
entry nvpp1 {
if match all {
ethernet-destination-address 00:04:96:00:00:00 / ff:ff:ff:00:00:00 ;
} then {
deny ;
count host1
} }
```

The following is the policy2.pol file for Port 21 in the egress direction:

```
entry nevpp1 {
if match all {
ethernet-source-address 00:04:96:00:00:00 / ff:ff:ff:00:00:00 ;
} then {
deny ;
count h1
} }
```

The following commands configure VM authentication in the local database:

```
create vm-tracking local-vm mac-address 00:04:96:27:C8:23
configure vm-tracking local-vm mac-address 00:04:96:27:C8:23 ip-address 11.1.1.101
configure vm-tracking local-vm mac-address 00:04:96:27:C8:23 name myVm1
create vm-tracking vpp vpp1
configure vm-tracking vpp vpp1 add ingress policy policy1
configure vm-tracking vpp vpp1 add egress policy policy2
configure vm-tracking local-vm mac-address 00:04:96:27:C8:23 vpp vpp1
```

The following commands used to create VM-mac with vlan-tag, and Vr for Dynamic vlan creation:

```
create vm-tracking local-vm mac-address 00:00:00:00:00:01
configure vm-tracking local-vm mac-address 00:00:00:00:00:01 vpp lvpp1
configure vm-tracking local-vm mac-address 00:00:00:00:00:01 vlan-tag 1000 vr VR-Default
configure vm-tracking vpp lvpp1 vlan-tag 2000
```

The following commands display the switch XNV feature status after configuration:

```
* Switch.67 # show vm-tracking local-vm

MAC Address        IP Address        Type     Value
-------------------------------------------------------------------------------
00:00:00:00:00:01                    VM
                                     VPP      lvpp1
                                     VLAN Tag 1000
                                     VR Name  VR-Default
Number of Local VMs: 1
* Switch.69 # show vm-tracking vpp

VPP Name                          Type       Value
-------------------------------------------------------------------------------
lvpp1                             origin     local
                                  counters   none
                                  VLAN Tag   2000
                                  VR Name Vr-Default
ingress             policy1
egress              policy2
Number of Local VPPs : 1
Number of Network VPPs: 0
Switch.71 # show vm-tracking
-------------------------------------------------------------
   VM Tracking Global Configuration
-------------------------------------------------------------
VM Tracking                     : Enabled
VM Tracking authentication order: nms vm-map local
VM Tracking nms reauth period   : 0 (Re-authentication disabled)
VM Tracking blackhole policy    : none
-------------------------------------------------------------
```

```
Port                          : 19
VM Tracking                   : Enabled
VM Tracking Dynamic VLAN      : Enabled

                    Flags
MAC                 APC   IP Address      Type     Value
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------

Flags :
  (A)uthenticated     : L - Local, N - NMS, V - VMMAP
  (P)olicy Applied    : B - All Ingress and Egress, E - All Egress, I - All Ingress
  (C)ounter Installed : B - Both Ingress and Egress, E - Egress Only, I - Ingress Only

Type :
  IEP - Ingress Error Policies
  EEP - Egress Error Policies

Number of Network VMs Authenticated: 0
Number of Local VMs Authenticated  : 0
Number of VMs Authenticated        : 0
Switch.73 # show policy
Policies at Policy Server:
PolicyName                   ClientUsage     Client          BindCount
-------------------------------------------------------------------------
policy1                           1              acl             1
policy2                           1              acl             1
```

Network (VMMAP) Authentication Configuration

If you only want to use network authentication, configure the *XNV*-enabled switches as follows:

```
configure vm-tracking authentication database-order vm-map
```

After the repository server is configured (see Repository Server Setup on page 539), the following commands can be used to display the switch XNV feature status:

```
* Switch.32 # show vm-tracking repository
---------------------------------
VMMAP FTP Server Information
---------------------------------
Primary VMMAP FTP Server  :
Server name:
IP Address      : 10.127.8.1
VR Name         : VR-Mgmt
Path Name       : /pub (default)
User Name       : anonymous (default)
Secondary VMMAP FTP Server : Unconfigured
Last sync           : 16:56:11          Last sync server : Primary
Last sync status    : Successful
* Switch.69 # show vm-tracking vpp
VPP Name                    Type               Name
-------------------------------------------------------------------------
nvpp1                       origin             network
ingress             nvpp1
egress              nevpp1
nvpp2                       origin             network
ingress             nvpp2
egress              nevpp2
```

```
Number of Local VPPs : 0
Number of Network VPPs: 2
* Switch.15 # show vm-tracking
-------------------------------------------------------------
VM Tracking Global Configuration
-------------------------------------------------------------
VM Tracking                     : Enabled
VM Tracking authentication order: vm-map
VM Tracking nms reauth period   : 0 (Re-authentication disabled)
VM Tracking blackhole policy    : none
-------------------------------------------------------------
Port                : 21
VM TRACKING         : ENABLED
Flags
MAC                AP    IP Address      Type     Name
-------------------------------------------------------------------------------------
00:04:96:27:c8:23  VB    11.1.1.101      VM       vm_1
VPP      nvpp1
00:04:96:27:c8:24  VB    11.1.1.102      VM       vm_2
VPP      nvpp2
-------------------------------------------------------------------------------------
Flags :
(A)uthenticated  : L - Local, N - NMS, V - VMMAP
(P)olicy Applied : B - Both, E - Egress, I - Ingress
Number of Network VMs Authenticated   : 2
Number of Local VMs Authenticated     : 0
Number of VMs Authenticated           : 2
* Switch.16 # show vm-tracking network-vm
MAC Address        IP Address        Type     Name
-------------------------------------------------------------------------------
00:04:96:27:c8:23  11.1.1.101        VM       vm_1
VPP      nvpp1
00:04:96:27:c8:23  11.1.1.102        VM       vm_2
VPP      nvpp2
Number of Network VMs: 2
* Switch.16 # show policy
Policies at Policy Server:
PolicyName                   ClientUsage     Client          BindCount
-------------------------------------------------------------------------
vmt/nvpp1                          1          acl             1
vmt/nvpp2                          1          acl             1
vmt/nevpp1                         1          acl             1
vmt/nevpp2                         1          acl             1

show vm-tracking nms server
VM Tracking NMS : enabled
VM Tracking NMS server connect time out: 3 seconds

Primary VM Tracking NMS server:
  Server name :
  IP address : 10.127.6.202
  Server IP Port: 1812
  Client address: 10.127.11.101 (VR-Mgmt)
  Shared secret : qijxou
Access Requests : 0 Access Accepts : 0
Access Rejects : 0 Access Challenges : 0
Access Retransmits: 0 Client timeouts : 0
Bad authenticators: 0 Unknown types : 0
Round Trip Time : 0
```

NMS Server Authentication Configuration

- If you only want to use NMS server authentication, configure the *XNV*-enabled switches as follows:

  ```
  configure vm-tracking authentication database-order nms
  ```

- Configure the NMS server as follows:
  a. Add Switch1 and Switch2 as *RADIUS* clients.
  b. Add the MAC addresses for VM1 and VM2 as users, and configure the passwords to match the user names.
  c. Add a remote access policy with the Extreme Networks VSAs:
     - Vendor code: 1916
     - VSA ID: 213 (EXTREME_VM_NAME)

       Example: MyVM1
     - VSA ID: 214 (EXTREME_VM_VPP_NAME)

       Example: nvpp1
     - VSA ID: 215 (EXTREME_VM_IP_ADDR)

       Example: 11.1.1.254

  Note
  For the Dynamic *VLAN* feature, the following VSAs are used:

  EXTREME_VM_VLAN_ID with VSA ID as 216
  EXTREME_VM_VR_NAME with VSA ID as 217

- The following command configures the switch as an NMS server client:

  ```
  configure vm-tracking nms primary server 10.127.5.221 client-ip 10.127.8.12 shared-secret secret
  ```

After the repository server is configured (see Repository Server Setup on page 539), the following commands can be used to display the switch XNV feature status:

```
 * Switch.33 # show vm-tracking nms server
VM Tracking NMS : enabled
VM Tracking NMS server connect time out: 3 seconds
Primary VM Tracking NMS server:
 Server name   :
 IP address    : 10.127.5.221
 Server IP Port: 1812
 Client address: 10.127.8.12 (VR-Mgmt)
 Shared secret : qijxou
 Access Requests   : 7               Access Accepts : 2
 Access Rejects    : 5               Access Challenges : 0
 Access Retransmits: 0               Client timeouts : 0
 Bad authenticators: 0               Unknown types : 0
 Round Trip Time   : 0

 * Switch.32 # show vm-tracking
-------------------------------------------------------------
VM Tracking Global Configuration
-------------------------------------------------------------
 VM Tracking                     : Enabled
 VM Tracking authentication order: nms
 VM Tracking nms reauth period   : 0 (Re-authentication disabled)
```

```
 VM Tracking blackhole policy    : none
-------------------------------------------------------------
 Port                        : 21
 VM TRACKING                 : ENABLED
 Flags
 MAC                AP    IP Address      Type     Name
-------------------------------------------------------------------------------
 00:04:96:27:c8:23  VB    11.1.1.101      VM       vm_1
 VPP      nvpp1
 00:04:96:27:c8:24  VB    11.1.1.102      VM       vm_2
 VPP      nvpp2
-------------------------------------------------------------------------------
 Flags :
 (A)uthenticated  : L - Local, N - NMS, V - VMMAP
 (P)olicy Applied : B - Both, E - Egress, I - Ingress
 Number of Network VMs Authenticated: 1
 Number of Local VMs Authenticated : 0
 Number of VMs Authenticated : 1

 * Switch.32 # show policy
Policies at Policy Server:
PolicyName ClientUsage Client BindCount
------------------------------------------------------------------------
vmt/nvpp1 1 acl 1
vmt/nvpp2 1 acl 1
------------------------------------------------------------------------
```

Managing Direct Attach to Support VEPA

You should only enable the Direct Attach feature on ports that directly connect to a VM server running VEPA software.

- To enable or disable the direct attach feature on a port, enter the command:

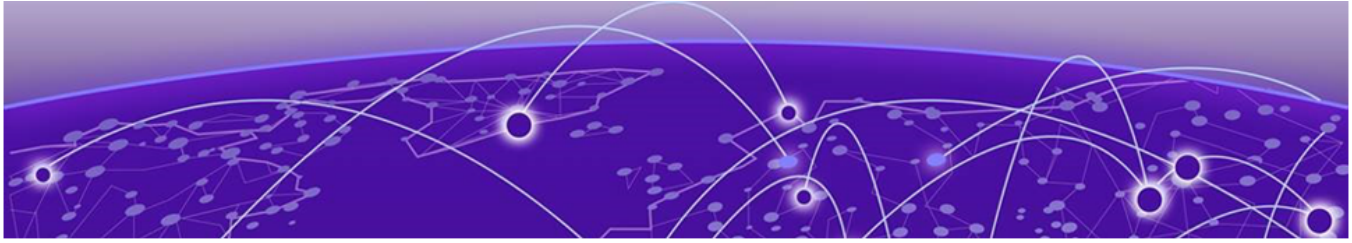
  `configure port` *`port`* **`reflective-relay`** [**`on`** | **`off`**]

- To see if the direct attach feature (reflective-relay) is enabled on a switch port, enter the command:

  `show ports information detail`

Note

When the Direct Attach feature is configured on a port, the port number cannot be included in the port list for a static *FDB* entry. For example, the Direct-Attach enabled port can be the only port number specified in a static FDB entry, but it cannot be included in a port-list range for a static FDB entry.



EAPS

EAPS Protocol Overview on page 548
Configuring EAPS on page 561
Displaying EAPS Information on page 571
Configuration Examples on page 572

This chapter provides an overview and discusses various topologies of feature. The chapter offers configuration and monitoring details, and also provides configuration examples.

EAPS Protocol Overview

The *EAPS (Extreme Automatic Protection Switching)* protocol provides fast protection switching to Layer 2 switches interconnected in an Ethernet ring topology, such as a Metropolitan Area Network (MAN) or large campus (see Figure 39).

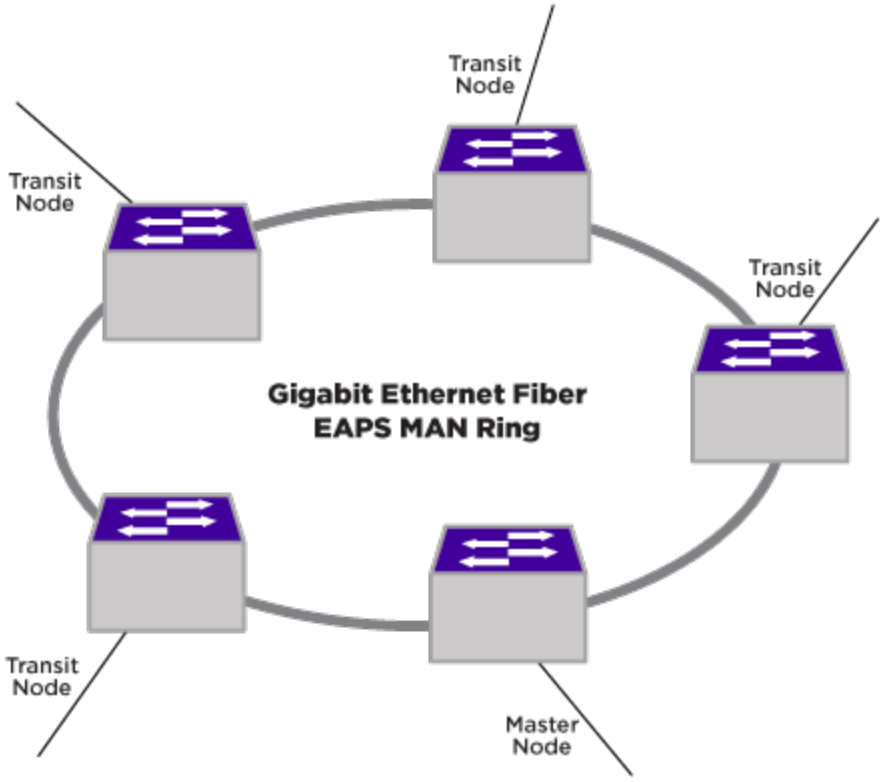


Figure 39: Gigabit Ethernet Fiber EAPS MAN Ring

EAPS Benefits

EAPS offers the following benefits:

- **Fast Recovery time for link or node failures**—When a link failure or switch failure occurs, EAPS provides fast recovery times. EAPS provides resiliency for voice, video and data services.
- **Scalable network segmentation and fault isolation**—EAPS domains can protect groups of multiple VLANs, allowing scalable growth and broadcast loop protection. EAPS domains provide logical and physical segmentation, which means the failures in one EAPS ring do not impact network service for other rings and VLANs.
- **Resilient foundation for non-stop IP routing services**—EAPS provides a resilient foundation for upper level routing protocols such as *OSPF (Open Shortest Path First)* and *BGP (Border Gateway Protocol)*, minimizing route-flapping and dropped neighbors within the routed IP network.
- **Predictable convergence regardless of failure location**—EAPS provides consistent and predictable recovery behavior regardless of where link failures occur. The simple blocking architecture and predictable performance of EAPS allows for enforceable Service Level Agreements (SLAs). This allows easier network troubleshooting and failure scenario analysis without lengthy testing or debugging on live production networks.

EAPS protection switching is similar to what can be achieved with the *STP (Spanning Tree Protocol)*, but EAPS offers the advantage of converging in less than one second when a link in the ring breaks.

An Ethernet ring built using EAPS can have resilience comparable to that provided by SONET rings, at a lower cost and with fewer restraints (such as ring size). The EAPS technology developed by Extreme Networks to increase the availability and robustness of Ethernet rings is described in *RFC 3619: Extreme Networks' Ethernet Automatic Protection Switching (EAPS) Version 1*.

EAPS Single Ring Topology

The simplest *EAPS* configuration operates on a single ring.

This section describes how this type of EAPS configuration operates. Later sections describe more complex configurations.

An EAPS domain consists of one master node and one or more transit nodes (see Figure 40), and includes one control *VLAN (Virtual LAN)* and one or more protected VLANs.

A domain is a single instance of the EAPS protocol that defines the scope of protocol operation. A single logical EAPS domain typically exists on a given physical ring topology (fiber or copper).

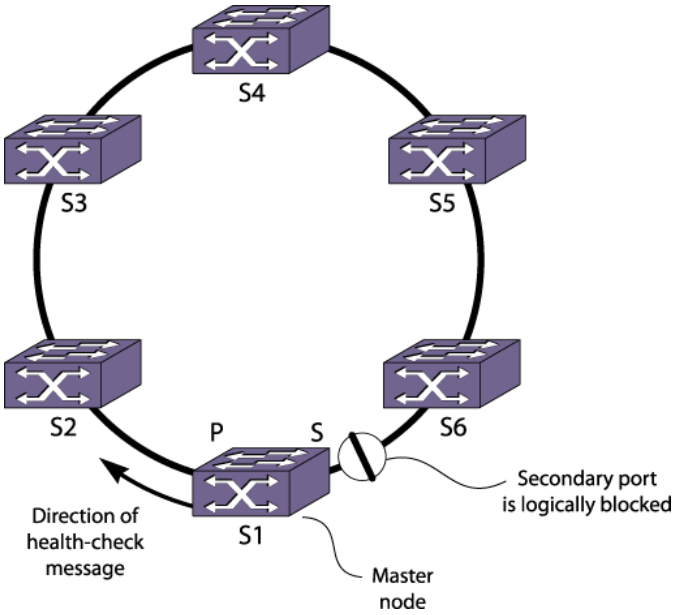


Figure 40: EAPS Operation

A protected VLAN is a user data VLAN that uses the ring for a protected connection between all edge ports. The protected VLAN uses 802.1q trunking on the ring ports and supports tagged and untagged edge ports.

One ring port of the master node is designated the master node's primary port (P), and another port is designated as the master node's secondary port (S) to the ring. In normal operation, the master node blocks the secondary port for all protected VLAN

traffic, thereby preventing a loop in the ring. (The spanning tree protocol, *STP*, provides the same type of protection.) Traditional Ethernet bridge learning and forwarding database mechanisms direct user data around the ring within the protected VLANs.

Note
Although primary and secondary ports are configured on transit nodes, both port types operate identically as long as the transit node remains a transit node. If the transit node is reconfigured as a master node, the configured states of the primary and secondary ports apply.

The control VLAN is a dedicated 802.1q tagged VLAN that is used to transmit and receive EAPS control frames on the ring. The control VLAN can contain only two EAPS ring ports on each node. Each EAPS domain has a unique control VLAN, and control traffic is not blocked by the master node at any time. The control VLAN carries the following EAPS control messages around the ring:

- Health-check messages, which are sent from the master node primary port. Transit nodes forward health-check messages toward the master node secondary port on the control VLAN. When the master node receives a health check message on the secondary port, the EAPS ring is considered intact.
- Link-down alert messages, which are sent from a transit node to the master node when the transit node detects a local link failure.
- Flush-*FDB (forwarding database)* messages, which are sent by the master node to all transit nodes when ring topology changes occur. Upon receiving this control frame, the transit node clears its MAC address forwarding table (FDB) and relearns the ring topology.

When the master node detects a failure, due to an absence of health-check messages or a received link-down alert, it transitions the EAPS domain to the Failed state and unblocks its secondary port to allow data connectivity in the protected VLANs.

EAPS Multiple Ring Topology

EAPS works with multiple ring networks to support more complex topologies for interconnecting multiple EAPS domains. This allows larger EAPS end-to-end networks to be built from edge to core.

Note
Minimal EAPS support is provided at all license levels. EAPS multiple ring topologies and common link topologies are supported at higher license levels as described in the Switch Engine v33.6.1 Licensing Guide document.

The simplest multiple ring topology uses a single switch to join two EAPS rings.

The common link feature uses two switches, which share a common link, to provide redundancy and link multiple EAPS rings.

Two Rings Connected by One Switch

Figure 41 shows how a data *VLAN* can span two rings interconnected by a common switch—a figure eight topology.

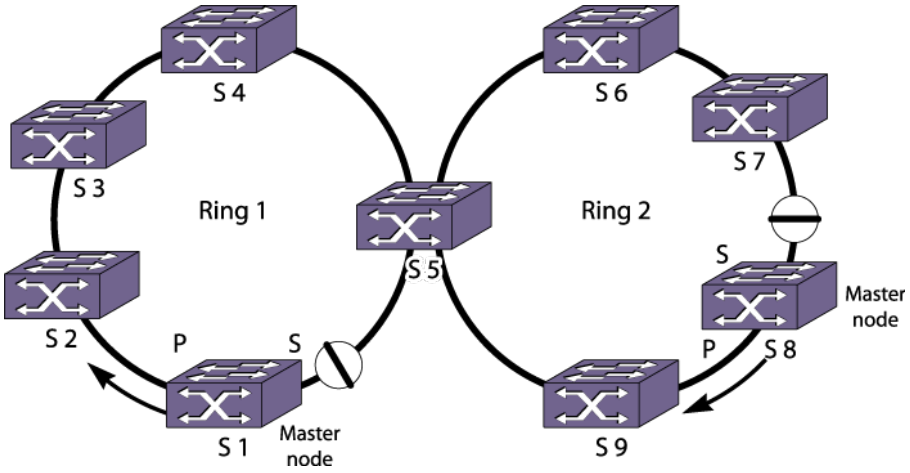


Figure 41: Two Rings Interconnected by One Switch

A data VLAN that spans multiple physical rings or *EAPS* domains and is protected by EAPS is called an overlapping VLAN. An overlapping VLAN requires loop protection for each EAPS domain to which it belongs.

In this example, there is an EAPS domain with its own control VLAN running on ring 1 and another EAPS domain with its own control VLAN running on ring 2. A data VLAN that spans both rings is added as a protected VLAN to both EAPS domains to create an overlapping VLAN. Switch S5 has two instances of EAPS domains running on it, one for each ring.

Multiple Rings Sharing an EAPS Common Link

EAPS Common Link Operation

Figure 42 shows an example of a multiple ring topology that uses the *EAPS* common link feature to provide redundancy for the switches that connect the rings.

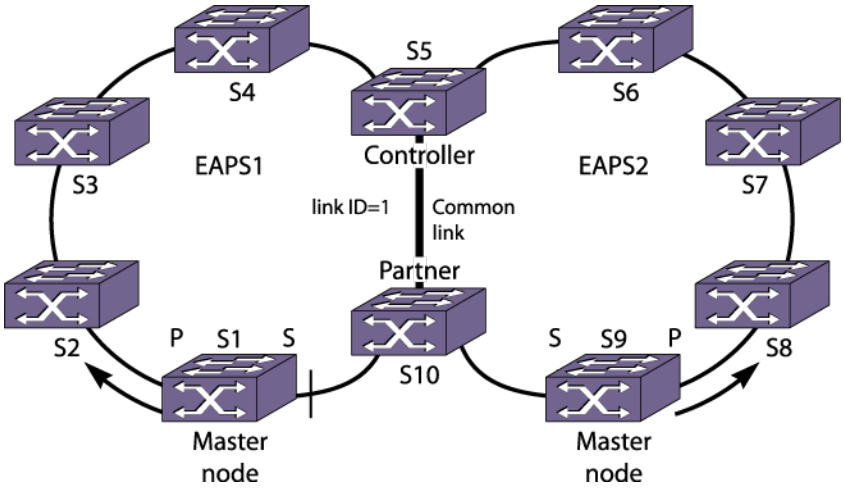


Figure 42: Multiple Rings Sharing a Common Link

An EAPS common link is a physical link that carries overlapping *VLAN*s that are protected by more than one EAPS domain.

In the example shown earlier, switch S5 could be a single point of failure. If switch S5 were to go down, users on Ring 1 would not be able to communicate with users on Ring 2. To make the network more resilient, you can add another switch. A second switch, S10, connects to both rings and to S5 through a common link, which is common to both rings.

The EAPS common link requires special configuration to prevent a loop that spans both rings. The software entity that requires configuration is the eaps shared-port, so the common link feature is sometimes called the *shared port* feature.

Note
If the shared port is not configured and the common link goes down, a superloop between the multiple EAPS domains occurs.

The correct EAPS common link configuration requires an EAPS shared port at each end of the common link. The role of the shared port (and switch) at each end of the common link must be configured as either controller or partner. Each common link requires one controller and one partner for each EAPS domain. Typically the controller and partner nodes are distribution or core switches. A controller or partner can also perform the role of master or transit node within its EAPS domain.

During normal operation, the master node on each ring protects the ring as described in EAPS Single Ring Topology on page 550. The controller and partner nodes work together to protect the overlapping VLANs from problems caused by a common link failure or a failed controller (see Figure 43).

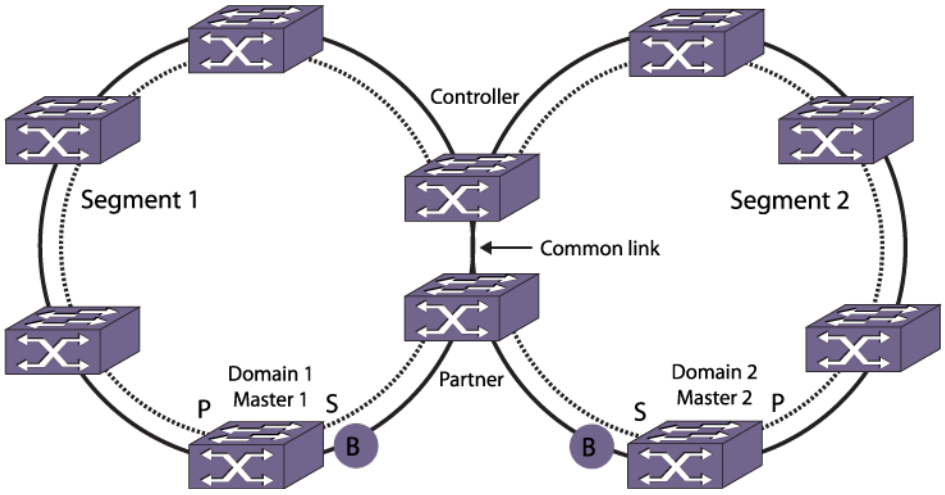


Figure 43: Master Node Operation in a Multiple Ring Topology

If a link failure occurs in one of the outer rings, only a single EAPS domain is affected. The EAPS master detects the failure in its domain, and converges around the failure. In this case, the controller does not take any blocking action, and EAPS domains on other rings are not affected. Likewise, when the link is restored, only the local EAPS domain is affected. The controller and any EAPS domains on other rings are not affected, and continue forwarding traffic normally.

To detect common-link faults, the controller and partner nodes send segment health check messages at one-second intervals to each other through each segment. A

segment is the ring communication path between the controller and partner. The common link completes the ring, but it is a separate entity from the segment. To discover segments and their up or down status, segment health-check messages are sent from controller to partner, and also from partner to controller (see Figure 44).

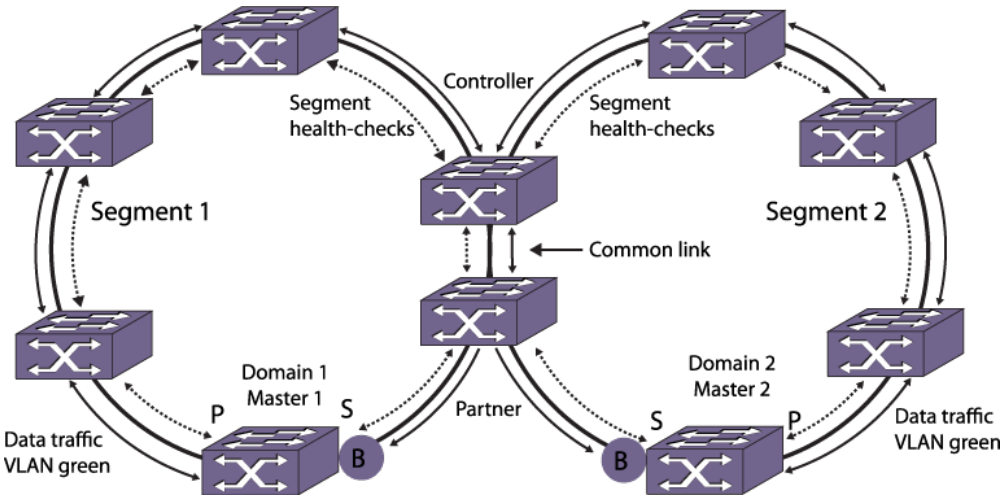


Figure 44: Segment Health-Check Messages

Common Link Fault Detection and Response

With one exception, when a common link fails, each master node detects the failure and unblocks its secondary port, as shown in Figure 45.

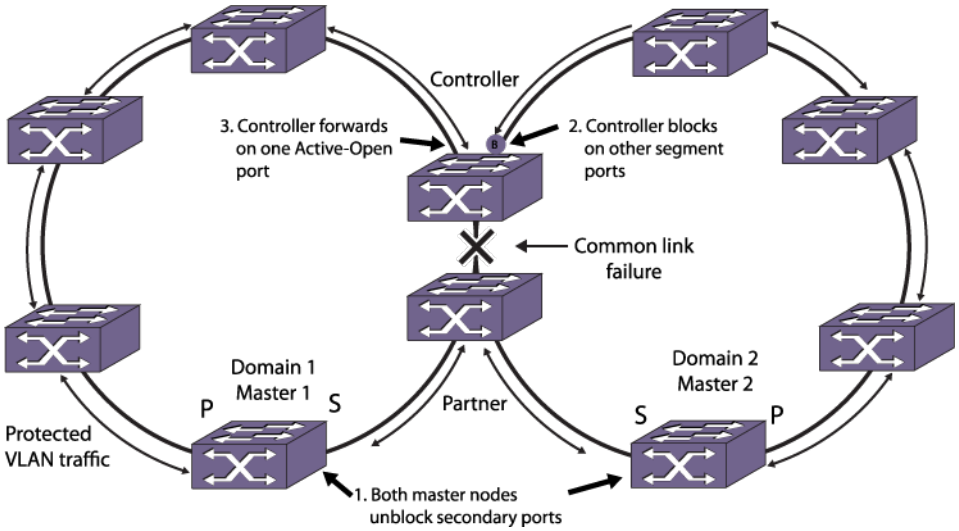


Figure 45: Common Link Failure

Because the secondary port of each master node is now unblocked, the new topology introduces a broadcast loop spanning the outer rings.

The controller and partner nodes immediately detect the loop, and the controller does the following:

- Selects an active-open port for protected VLAN communications. The active-open port is the port with the smallest segment-port among all segments that are in “up” state.
- Blocks protected VLAN communications on all segment ports except the active-open port.

Note
When a controller goes into or out of the blocking state, the controller sends a flush-fdb message to flush the *FDB* in each of the switches in its segments. In a network with multiple EAPS ports in the blocking state, the flush-fdb message gets propagated across the boundaries of the EAPS domains.

Note
If CFM is not used, it is recommended to have a direct Link connected between EAPS Controller Node and EAPS Partner Node.

The exception mentioned above occurs when the partner node is also a master node, and the shared port that fails is configured as a primary port. In this situation, the master node waits for a link-down PDU from the controller node before opening the secondary port. This delay prevents a loop that might otherwise develop if the master/partner node detects the link failure before the controller node.

Note
If the common link and a ring link fail, and if the common link restores before the ring link, traffic down time can be as long as three seconds. This extended delay is required to prevent loops during the recovery of multiple failed links.

Common Link Recovery

When a common link recovers, each master node detects that the ring is complete and immediately blocks their secondary ports. The controller also detects the recovery and puts its shared port to the common link into a temporary blocking state called pre-forwarding as shown in Figure 46.

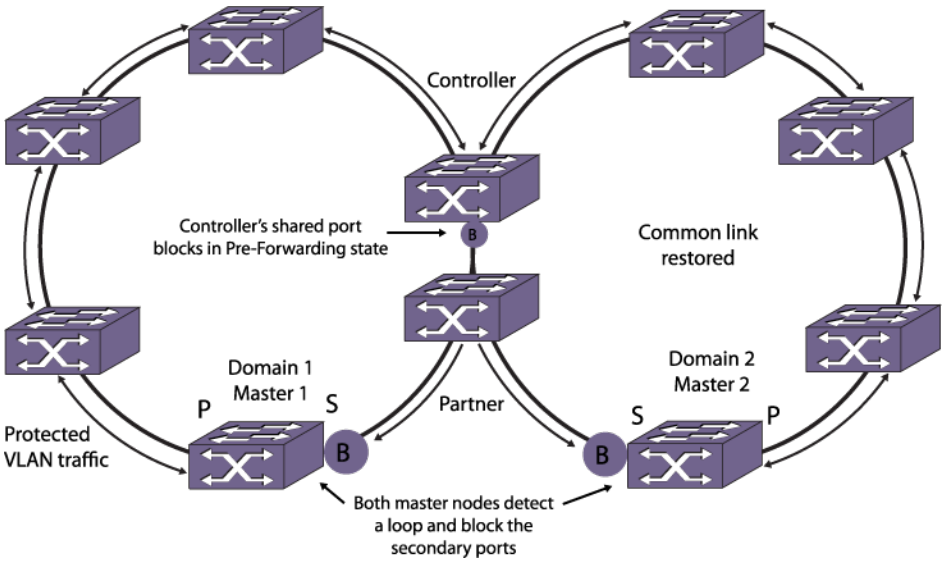


Figure 46: Common Link in Pre-Forwarding State

Because the topology has changed, the EAPS nodes must learn the new traffic paths. Each master node notifies all switches in their domain to clear their *FDB* tables, and traditional Ethernet bridge learning and forwarding mechanisms establish the new traffic paths. Once the controller receives flush-fdb messages for all of its connected EAPS domains, the controller shared-port state for the common link changes to forwarding, the controller state changes to Ready, and traffic flows normally as shown in Figure 47.

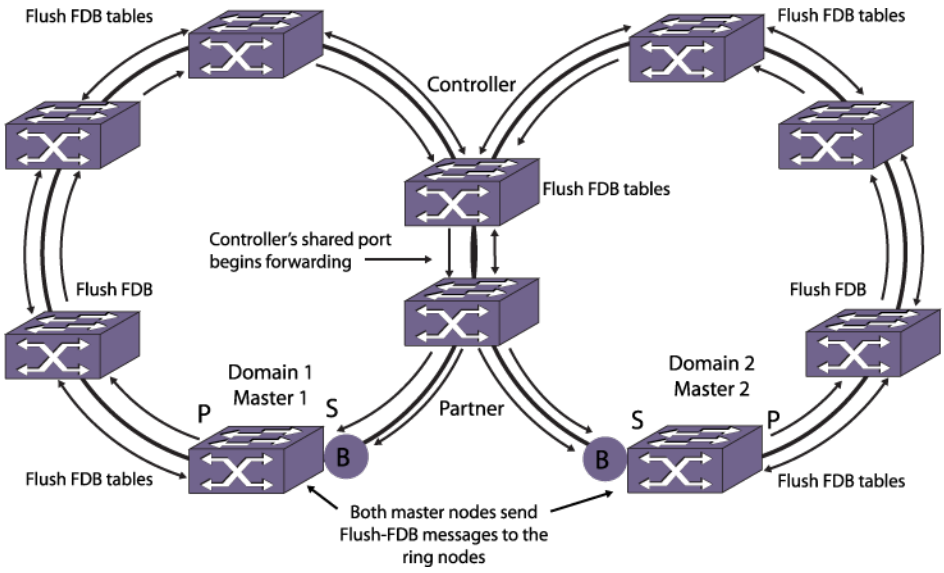


Figure 47: Common-Link Restored

Controller and Partner Node States

EAPS controller and partner nodes can be in the following states:

- **Ready**—Indicates that the EAPS domains are running, the common-link neighbor can be reached through segment health-checks, and the common link is up.
- **Blocking**—Indicates that the EAPS domains are running, the common-link neighbor can be reached through segment health-checks, but the common-link is down. Only the controller node (and not the partner) performs blocking.
- **Preforwarding**—Indicates the EAPS domain was in a blocking state, and the common link was restored. The controller port is temporarily blocked to prevent a loop during state transition from Blocking to Ready.
- **Idle**—Indicates the EAPS common-link neighbor cannot be reached through segment health-check messages.

Spatial Reuse with an EAPS Common Link

The common-link topology supports multiple *EAPS* domains (spatial reuse) on each ring as shown in Figure 48.

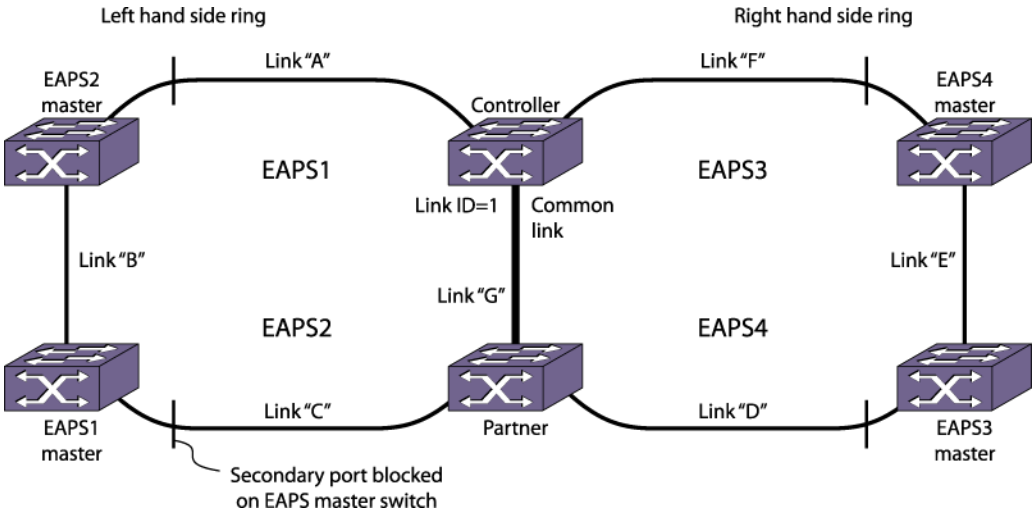


Figure 48: EAPS Common Link Topology with Spatial Reuse

Note
If you are using the older method of enabling *STP* instead of EAPSv2 to block the super loop in a shared-port environment, you can continue to do so. In all other scenarios, we recommend that you do not use both STP and EAPS on the same port.

Additional Common Link Topology Examples

Basic Core Topology

Figure 49 shows a core topology with two access rings. In this topology, there are two *EAPS* common links.

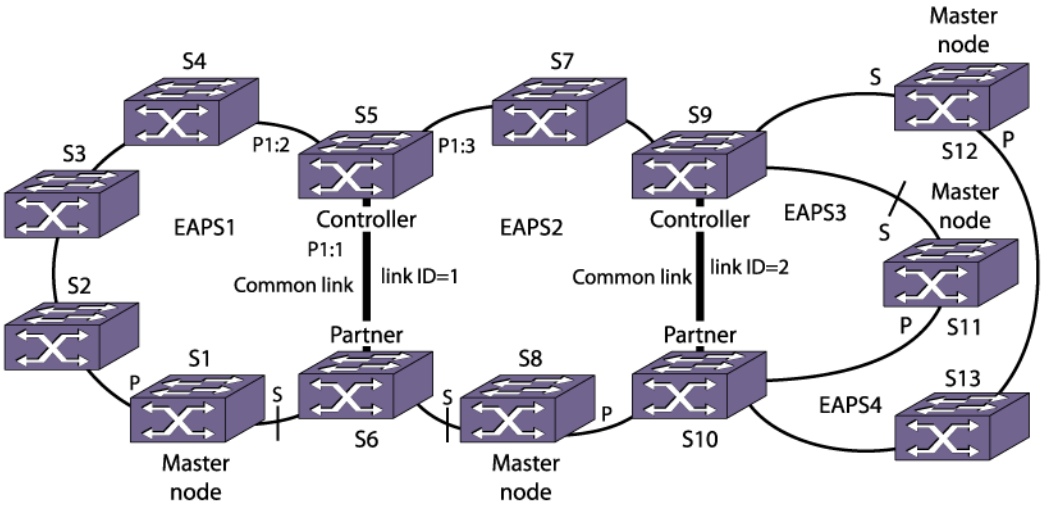


Figure 49: Basic Core Topology

Right-Angle Topology

In the right-angle topology, there are still two EAPS common links, but the common links are adjacent to each other.

To configure a right-angle topology, there must be two common links configured on one of the switches. Figure 50 shows a right-angle topology.

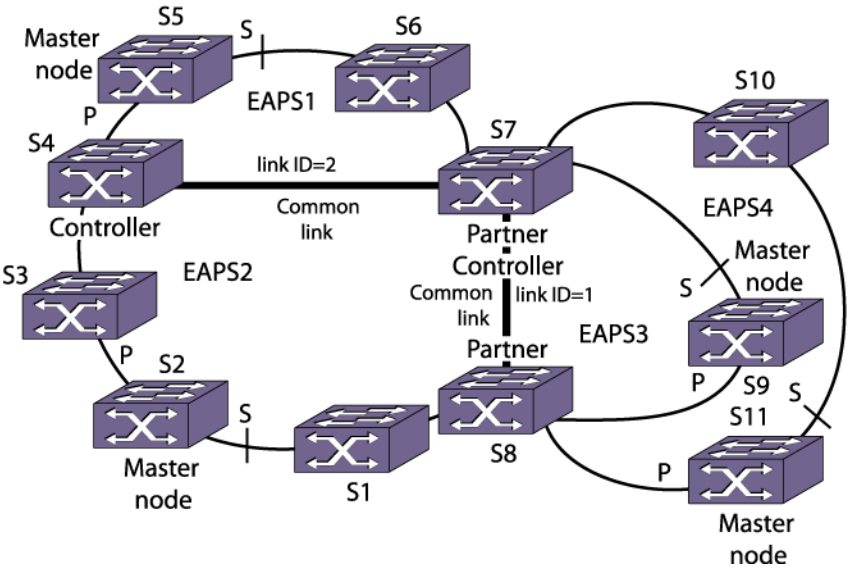


Figure 50: Right-Angle Topology

Combined Basic Core and Right-Angle Topology

Figure 51 shows a combination basic core and right-angle topology.

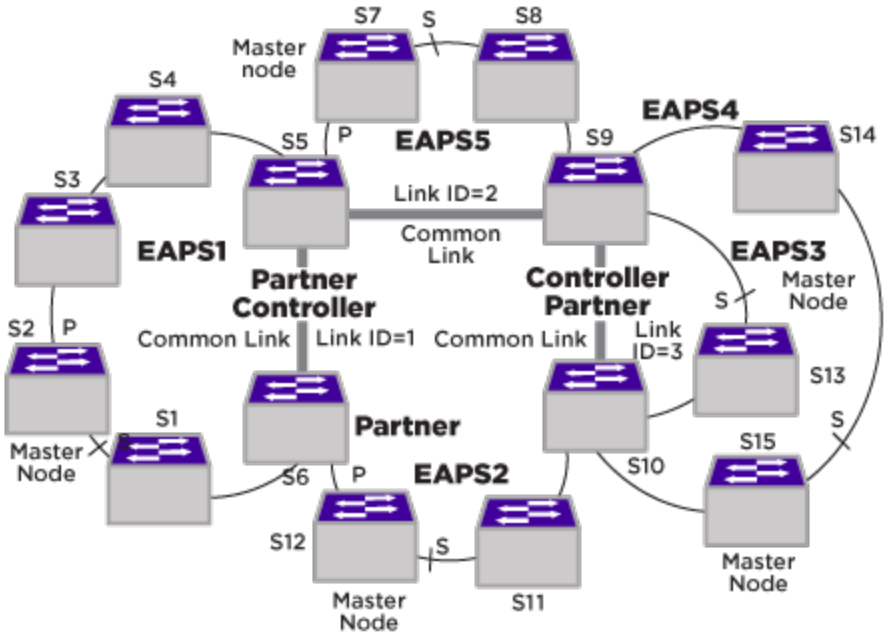


Figure 51: Basic Core and Right Angle Topology

Figure 52 shows an extension of the basic core and right angle configuration.

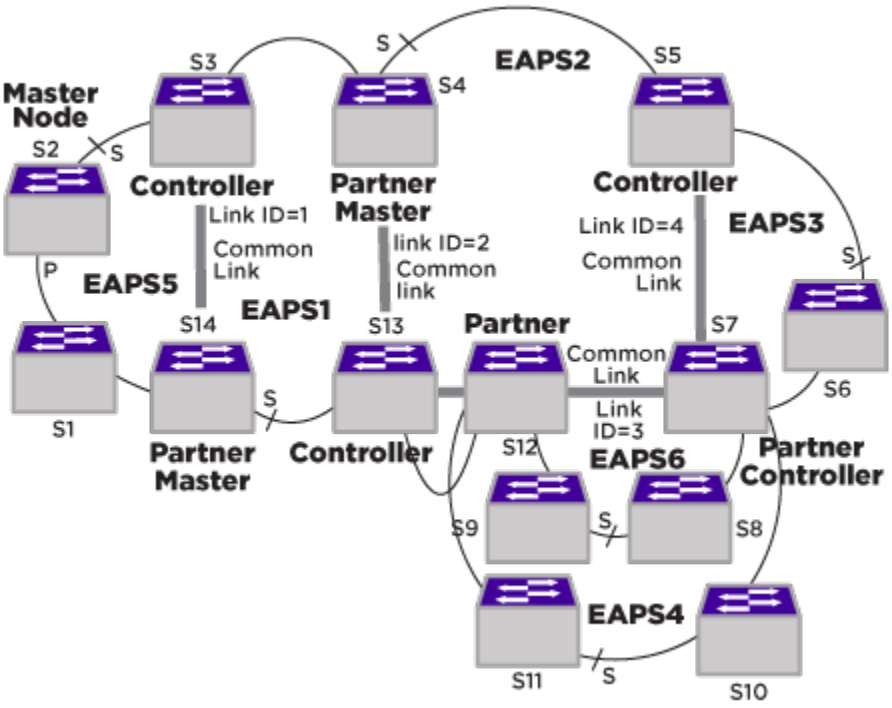


Figure 52: Advanced Basic Core and Right Angle Topology

Large Core and Access Ring Topology

Figure 53 shows a single large core ring with multiple access rings hanging off of it. This is an extension of a basic core configuration.

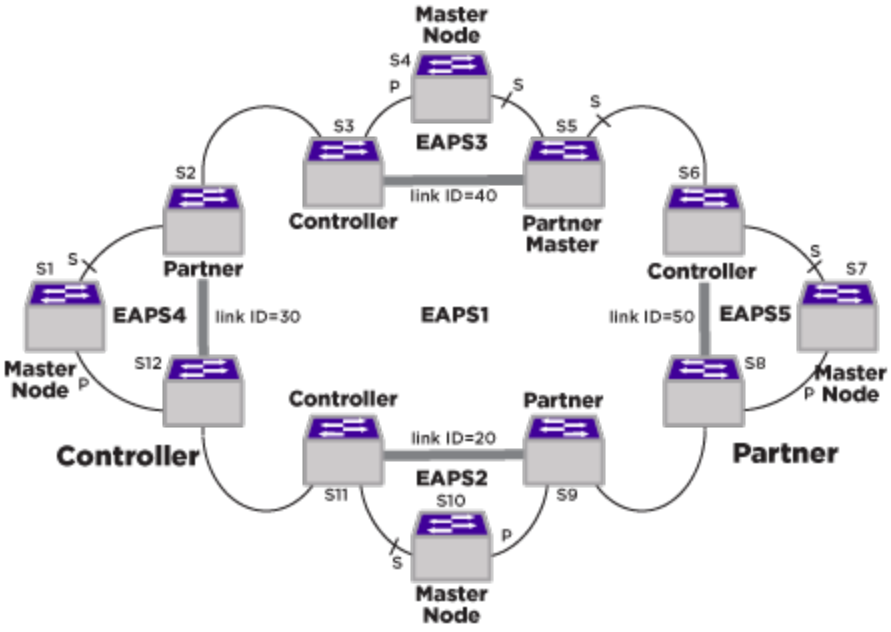


Figure 53: Large Core and Access Ring Topology

fast convergence

In EAPS, Fast Convergence allows convergence in the range of 50 milliseconds. This parameter is configured for the entire switch, not by EAPS domain.

EAPS and Hitless Failover—SummitStack Only

When you use redundancy in a SummitStack, one node assumes the role of primary and another node assumes the role of backup.

The primary node executes the switch's management functions, and the backup node acts in a standby role. Hitless failover transfers switch management control from the primary to the backup and maintains the state of *EAPS*. EAPS supports hitless failover. You do not explicitly configure hitless failover support; rather, if you are operating with redundancy in a SummitStack, hitless failover is available.

Note

Not all platforms support hitless failover in the same software release. To verify if the software version you are running supports hitless failover, see the table in Protocol Support for Hitless Failover on page 90. For more information about protocol and platform support for hitless failover, see Understanding Hitless Failover Support on page 89.

To support hitless failover, the primary node replicates all EAPS PDUs to the backup, which allows the backup to be aware of the EAPS domain state. Since both nodes receive EAPS PDUs, each node maintains equivalent EAPS states.

By knowing the state of the EAPS domain, the EAPS process running on the backup node can quickly recover after a primary node failover. Although both nodes receive EAPS PDUs, only the primary transmits EAPS PDUs to neighboring switches and actively participates in EAPS.

Note
For instructions on how to manually initiate hitless failover, see Relinquishing Primary Status on page 86.

EAPS Licensing

Different *EAPS* features are offered at different license levels.

For complete information about software licensing, including how to obtain and upgrade your license and what licenses are appropriate for these features, in the Switch Engine v33.6.1 Licensing Guide document.

Configuring EAPS

Single Ring Configuration Tasks

To configure and enable an *EAPS* protected ring, do the following on each ring node:

1. Create an EAPS domain and assign a name to the domain as described in Creating and Deleting an EAPS Domain on page 562.
2. Create and add the control *VLAN* to the domain as described in Adding the EAPS Control VLAN on page 562.
3. Create and add the protected VLAN(s) to the domain as described in Adding Protected VLANs on page 562.
4. Configure the EAPS mode (master or transit) for the switch in the domain as described in Defining the Switch Mode (Master or Transit) on page 563.
5. Configure the EAPS ring ports, including the master primary and secondary ring ports, as described in Configuring the Ring Ports on page 563.
6. If desired, configure the polling timers and timeout action as described in Configuring the Polling Timers and Timeout Action on page 564.
7. Enable EAPS for the entire switch as described in Enabling and Disabling EAPS on the Switch on page 565.

8. If desired, enable Fast Convergence as described in Enabling and Disabling Fast Convergence on page 566.
9. Enable EAPS for the specified domain as described in Enabling and Disabling an EAPS Domain on page 566.

Note
If you configure a VMAN on a switch running EAPS, make sure you configure the VMAN attributes on all of the switches that participate in the EAPS domain. For more information about VMANs, see VMAN (PBN) and PBBN.

Creating and Deleting an EAPS Domain

Each *EAPS* domain is identified by a unique domain name.

- To create an EAPS domain, use the following command:

  ```
  create eaps name
  ```

- To delete an EAPS domain, use the following command:

  ```
  delete eaps name
  ```

Adding the EAPS Control VLAN

You must create and configure one control *VLAN* for each *EAPS* domain. For instructions on creating a VLAN, see VLANs on page 2213.

- To configure EAPS to use a VLAN as the EAPS control VLAN for a domain, use the following command:

  ```
  configure eaps name add control {vlan} vlan_name
  ```

  Note
  A control VLAN cannot belong to more than one EAPS domain. If the domain is active, you cannot delete the domain or modify the configuration of the control VLAN.
  The control VLAN must NOT be configured with an IP address. In addition, only ring ports may be added to this control VLAN. No other ports can be members of this VLAN. Failure to observe these restrictions can result in a loop in the network.
  The ring ports of the control VLAN must be tagged.

  By default, EAPS PDUs are automatically assigned to *QoS (Quality of Service)* profile QP8. This ensures that the control VLAN messages reach their intended destinations. You do not need to configure a QoS profile for the control VLAN.

Adding Protected VLANs

You must add one or more protected *VLAN*s to each *EAPS* domain. The protected VLANs are the data-carrying VLANs.

Note
When you configure a protected VLAN, the ring ports of the protected VLAN must be tagged (except in the case of the default VLAN).

For instructions on creating a VLAN, see VLANs on page 2213.

- To configure a VLAN as an EAPS protected VLAN, use the following command:

  ```
  configure eaps name add protected {vlan} vlan_name
  ```

Configuring the EAPS Domain Priority

The *EAPS* domain priority feature allows you to select the EAPS domains that are serviced first when a break occurs in an EAPS ring. For example, you might set up a network topology with two or more domains on the same physical ring, such as in spatial reuse. In this topology, you could configure one domain as high priority and the others as normal priority. You would then add a small subset of the total protected VLANs to the high priority domain, and add the rest of the protected vlans to the normal priority domain. The secondary port of the normal and high priority domains can be the same, or as is typically the case of spatial reuse, opposite. If a ring fault occurs in this topology, the protected VLANs in the high priority domain are the first to recover.

- To configure the EAPS domain priority, use the following command:

  ```
  configure eaps name priority {high | normal}
  ```

Defining the Switch Mode (Master or Transit)

We recommend keeping the loop protection warning messages enabled. If you have considerable knowledge and experience with *EAPS*, you might find the EAPS loop protection warning messages unnecessary.

1. Configure the EAPS switch mode for a domain using the following command:

   ```
   configure eaps name mode [master | transit]
   ```

   One switch on the ring must be configured as the master node for the specified domain; all other switches on the same ring and domain are configured as transit nodes.

   If you configure a switch to be a transit node for an EAPS domain, the default switch configuration displays the following message and prompts you to confirm the command:

   ```
   WARNING: Make sure this specific EAPS domain has a Master node in the ring.  If you
   change this node from EAPS master to EAPS transit, you could cause a loop in the
   network. Are you sure you want to change mode to transit? (y/n)
   ```

2. When prompted, do one of the following:
   - Enter `y` to identify the switch as a transit node.
   - Enter `n` or press **[Return]** to cancel the command.

For more information see, Disabling EAPS Loop Protection Warning Messages on page 567.

Configuring the Ring Ports

Each node on the ring connects to the ring through two ring ports. The ports that you choose on each switch should be tagged and added to the control *VLAN* and all protected VLANs. For information on adding tagged ports to a VLAN, see VLANs on page 2213.

On the master node, one ring port must be configured as the primary port, and the other must be configured as the secondary port.

We recommend that you keep the loop protection warning messages enabled. If you have considerable knowledge and experience with *EAPS*, you might find the EAPS loop protection warning messages unnecessary.

1. To configure a node port as primary or secondary, use the following command:
   `configure eaps` *`name`* [**`primary`** | **`secondary`**] **`port`** *`ports`*

   If you attempt to add an EAPS ring port to a VLAN that is not protected by EAPS, the default switch configuration prompts you to confirm the command with the following message:
   ```
   Make sure <vlan_name> is protected by EAPS. Adding EAPS ring ports to a VLAN could
   cause a loop in the network. Do you really want to add these ports (y/n)
   ```
2. When prompted, do one of the following:
   - Enter `y` to identify the switch as a transit node.
   - Enter `n` or press **[Return]** to cancel the command.

For information on configuring a VLAN for EAPS, see the following sections:

- Adding the EAPS Control VLAN on page 562
- Adding Protected VLANs on page 562

For more information see, Disabling EAPS Loop Protection Warning Messages on page 567.

Configuring the Polling Timers and Timeout Action

The polling timers provide an alternate way to detect ring breaks. In a ring that uses only Extreme Networks switches, the master switch learns about a ring break by receiving a link-down PDU. When the ring uses only Extreme networks switches, the polling timers are not needed and can remain configured for the default values.

In a ring that contains switches made by other companies, the polling timers provide an alternate way to detect ring breaks. The master periodically sends hello PDUs at intervals determined by the hello PDU timer and waits for a reply. If a hello PDU reply is not received before the failtime timer expires, the switch detects a failure and responds by either sending an alert or opening the secondary port. The response action is defined by a configuration command.

- Set the polling timer values the master node uses for detecting ring failures.
  `configure eaps` *`name`* **`hellotime`** *`seconds milliseconds`*

  `configure eaps` *`name`* **`failtime`** *`seconds milliseconds`*

  Note
  These commands apply only to the master node. If you configure the polling timers for a transit node, they are ignored. If you later reconfigure that transit node as the master node, the polling timer values are used as the current values.

  Use the hellotime keyword and its associated parameters to specify the amount of time the master node waits between transmissions of health check messages on

the control *VLAN*. The combined value for seconds and milliseconds must be greater than 0. The default value is 1 second.

Use the failtime keyword and its associated parameters to specify the amount of time the master node waits before the failtimer expires. The combined value for seconds and milliseconds must be greater than the configured value for hellotime. The default value is 3 seconds.

Note
Increasing the failtime value increases the time it takes to detect a ring break using the polling timers, but it can also reduce the possibility of incorrectly declaring a failure when the network is congested.

- Configure the action taken when a ring break is detected.

  `configure eaps` *`name`* **`failtime expiry-action`** [**`open-secondary-port`** | **`send-alert`**]

  Use the *`send-alert`* parameter to send an alert when the failtimer expires. Instead of going into a failed state, the master node remains in a Complete or Init state, maintains the secondary port blocking, and writes a critical error message to syslog warning the user that there is a fault in the ring. An *SNMP (Simple Network Management Protocol)* trap is also sent.

  Use the *`open-secondary-port`* parameter to open the secondary port when the failtimer expires.

Enabling and Disabling EAPS on the Switch

We recommend that you keep the loop protection warning messages enabled. If you have considerable knowledge and experience with *EAPS*, you might find the EAPS loop protection warning messages unnecessary.

- To enable the EAPS function for the entire switch, use the following command:

  ```
  enable eaps
  ```

- To disable the EAPS function for the entire switch, use the following command:

  ```
  disable eaps
  ```

  If you enter the command to disable EAPS, the default switch configuration displays the following warning message and prompts you to confirm the command:

  ```
  WARNING: Disabling EAPS on the switch could cause a loop in the network! Are you sure you want to disable EAPS? (y/n)
  ```

- When prompted, do one of the following:

  a. Enter `y` to disable EAPS for the entire switch.

  b. Enter `n` or press **[Return]** to cancel the command.

For more information see, Disabling EAPS Loop Protection Warning Messages on page 567.

Enabling and Disabling Fast Convergence

You can enable or disable fast convergence for the entire switch to improve *EAPS* convergence times.

Note
Possible factors affecting EAPS fast convergence time:

- The medium type of the link being flapped (Fiber link-down events are detected faster than copper, causing better convergence).
- Number of VLANs protected by the EAPS domain (convergence time increases with the number of protected VLANs).
- Number of *FDB* entries present during the switch over (convergence time increases with the number of FDBs learned).
- Topology change event (link down or link up) causes the master node to send an FDB flush to all transits. In the event ofa shared port failure, FDB is flushed twice, causing an increase in convergence time.
- Number of hops between the switch where the link flap happens and the master node (convergence increases with the number of hops).

- To enable or disable fast convergence on the switch, use the following command:

  `configure eaps` **`fast-convergence`**`[`**`off`** `|` **`on`**`]`

Enabling and Disabling an EAPS Domain

We recommend that you keep the loop protection warning messages enabled. If you have considerable knowledge and experience with *EAPS*, you might find the EAPS loop protection warning messages unnecessary.

- To enable a specific EAPS domain, use the following command:

  `enable eaps {`*`name`*`}`

- To disable a specific EAPS domain, use the following command:

  `disable eaps {`*`name`*`}`

  If you enter the `disable eaps` command, the default switch configuration displays the following warning message and prompts you to confirm the command:

  ```
  WARNING: Disabling specific EAPS domain could cause a loop in the network! Are you sure
  you want to disable this specific EAPS domain? (y/n)
  ```

- When prompted, do one of the following:

  a. Enter `y` to disable EAPS for the specified domain.

  b. Enter `n` or press **[Return]** to cancel the command.

For more information see, Disabling EAPS Loop Protection Warning Messages on page 567.

Configuring EAPS Support for Multicast Traffic

The ExtremeXOS software provides several commands for configuring how *EAPS* supports multicast traffic after an EAPS topology change.

Note

EAPS multicast flooding must be enabled before the add-ring-ports feature will operate. For information on enabling EAPS multicast flooding, see the command:

```
configure eaps multicast temporary-flooding [on | off]
```

Unconfiguring an EAPS Ring Port

Unconfiguring an *EAPS* port sets its internal configuration state to INVALID, which causes the port to appear in the Idle state with a port status of Unknown. This occurs when you use the `show eaps {eapsDomain} {detail}` command to display the status information about the port.

We recommend that you keep the loop protection warning messages enabled. If you have considerable knowledge and experience with EAPS, you might find the EAPS loop protection warning messages unnecessary.

1. To unconfigure an EAPS primary or secondary ring port for an EAPS domain, use the following command:

   ```
   unconfigure eaps eapsDomain [primary | secondary] port
   ```

   To prevent loops in the network, the switch displays by default a warning message and prompts you to unconfigure the specified EAPS primary or secondary ring port.
2. When prompted, do one of the following:
   a. Enter `y` to unconfigure the specified port.
   b. Enter `n` or press **[Return]** to cancel this action.

   The following command example unconfigures this node's EAPS primary ring port on the domain "eaps_1":

   ```
   unconfigure eaps eaps_1 primary port
   WARNING: Unconfiguring the Primary port from the EAPS domain could cause a loop in
   The network! Are you sure you want to unconfigure the Primary EAPS Port? (y/n)
   ```
3. Enter `y` to continue and unconfigure the EAPS primary ring port. Enter `n` to cancel this action.

   The switch displays a similar warning message if you unconfigure the secondary EAPS port.

For more information see, Disabling EAPS Loop Protection Warning Messages on page 567.

Disabling EAPS Loop Protection Warning Messages

The switch displays by default loop protection messages when configuring the following *EAPS* parameters:

- Adding EAPS primary or secondary ring ports to a *VLAN*
- Deleting a protected VLAN

- Disabling the global EAPS setting on the switch
- Disabling an EAPS domain
- Configuring an EAPS domain as a transit node
- Unconfiguring EAPS primary or secondary ring ports from an EAPS domain

We recommend keeping the loop protection warning messages enabled. If you have considerable knowledge and experience with EAPS, you might find the EAPS loop protection warning messages unnecessary. For example, if you use a script to configure your EAPS settings, disabling the warning messages allows you to configure EAPS without replying to each interactive yes/no question.

- To disable loop protection messages, use the following command:

  ```
  configure eaps config-warnings off
  ```

- To re-enable loop protection messages, use the following command:

  ```
  configure eaps config-warnings on
  ```

Common Link Topology Configuration Tasks

To create a common link topology, you must configure the shared ports at each end of the common link.

EAPS Shared Port Configuration Rules

The following rules apply to *EAPS* shared port configurations:

- Each common link in the EAPS network must have a unique link ID, which is configured at the shared port at each end of the link.
- The shared port mode configured on each side of a common link must be different from the other; one must be a controller and one must be a partner.
- The controller and partner shared ports on either side of a common link must have the same
- link ID. The common link is established only when the shared ports at each end of the common link have the same link ID.
- There can be up to two shared ports per switch.
- There cannot be more than one controller on a switch.

  Valid combinations on any one switch are:

  - 1 controller
  - 1 partner

  - 1 controller and 1 partner
  - 2 partners
- A shared port cannot be configured on an EAPS master's secondary port.

Note
When a common link fails, one of the segment ports becomes the active-open port, and all other segment ports are blocked to prevent a loop for the protected VLANs. For some topologies, you can improve network performance during a common link failure by selecting the port numbers to which segments connect. For information on how the active-open port is selected, see Common Link Fault Detection and Response.

Common Link Configuration Overview

To configure and enable a common link to serve multiple rings, do the following on the controller and partner nodes:

1. Create a shared port for the common link as described in Creating and Deleting a Shared Port on page 569.
2. Configure the shared port as either a controller or a partner as described in Defining the Mode of the Shared Port on page 569.
3. Configure the link ID on the shared port as described in Configuring the Link ID of the Shared Port on page 570.
4. If desired, configure the polling timers and timeout action as described in Configuring the Shared Port Timers and Timeout Action on page 570.

   This step can be configured at any time, even after the *EAPS* domains are running.
5. Configure EAPS on each ring as described in Single Ring Configuration Tasks on page 561.

Creating and Deleting a Shared Port

To configure a common link, you must create a shared port on each switch belonging to the common link.

- To create a shared port, use the following command:

  ```
  create eaps shared-port ports
  ```

  Where *ports* is the common link port.

Note
A switch can have a maximum of two shared ports.

- To delete a shared port on the switch, use the following command:

  ```
  delete eaps shared-port ports
  ```

Defining the Mode of the Shared Port

The shared port on one end of the common link must be configured to be the controller. This is the end responsible for blocking ports when the common link fails, thereby preventing the superloop.

The shared port on the other end of the common link must be configured to be the partner. This end does not participate in any form of blocking. It is responsible for only sending and receiving health-check messages.

- To configure the mode of the shared port, use the following command:

  ```
  configure eaps shared-port ports mode controller | partner
  ```

Configuring the Link ID of the Shared Port

Each common link in the *EAPS* network must have a unique link ID. The controller and partner shared ports that belong to the same common link must have matching link IDs. No other instance in the network should have that link ID.

If you have multiple adjacent common links, we recommend that you configure the link IDs in ascending order of adjacency. For example, if you have an EAPS configuration with three adjacent common links, moving from left to right of the topology, configure the link IDs from the lowest to the highest value.

- To configure the link ID of the shared port, use the following command:

  ```
  configure eaps shared-port ports link-id id
  ```

  The link ID range is 1–65534.

Configuring the Shared Port Timers and Timeout Action

- To configure the shared port timers, use the following commands:

  ```
  configure eaps shared-port port common-path-timers {[health-interval | timeout] seconds}
  ```

  ```
  configure eaps shared-port port segment-timers health-interval seconds
  ```

  ```
  configure eaps shared-port port segment-timers timeout seconds
  ```

- To configure the time out action for segment timers, use the following command:

  ```
  configure eaps shared-port port segment-timers expiry-action [segment-down | send-alert]
  ```

Unconfiguring an EAPS Shared Port

- To unconfigure a link ID on a shared port, use the following command:

  ```
  unconfigure eaps shared-port ports link-id
  ```

- To unconfigure the mode on a shared port, use the following command:

  ```
  unconfigure eaps shared-port ports mode
  ```

- To delete a shared port, use the following command:

  ```
  delete eaps shared-port ports
  ```

Clearing the EAPS Counters

The *EAPS* counters continue to increment until you explicitly clear the information. By clearing the counters, you can see fresh statistics for the time period you are monitoring.

- To clear the counters used by EAPS, use the following commands:

  ```
  clear counters
  clear eaps counters
  ```

Displaying EAPS Information

Displaying Single Ring Status and Configuration Information

- To display *EAPS* status and configuration information, use the following command:

  ```
  show eaps {eapsDomain} {detail}
  ```

  Note
  You might see a slightly different display, depending on whether you enter the command on the master node or the transit node.

  If you specify a domain with the optional *eapsDomain* parameter, the command displays status information for a specific EAPS domain.

  The display from the `show eaps detail` command shows all the information shown in the `show eaps` *eapsDomain* command for all configured EAPS domains.

Displaying Domain Counter Information

- To display *EAPS* counter information for one or all domains, use the following command:

  ```
  show eaps counters [eapsDomain | global]
  ```

  If you specify the name of an EAPS domain, the switch displays counter information related to only that domain.

  If you specify the global keyword, the switch displays a list of the counter totals for all domains. To see the counters for a specific domain, you must specify the domain name.

  Note
  If a PDU is received, processed, and consumed, only the Rx counter increments. If a PDU is forwarded in slow path, both the Rx counter and Fw counter increment.

Displaying Common Link Status and Configuration Information

Each controller and partner node can display status and configuration information for the shared port or ports on the corresponding side of the common link.

- To display *EAPS* common link information, use the following command:

  `show eaps shared-port {port} {detail}`

  If you enter the `show eaps shared-port` command without an argument or keyword, the command displays a summary of status information for all configured EAPS shared ports on the switch.

  If you specify a shared port, the command displays information about that specific port.

  You can use the detail keyword to display more detailed status information about the segments and VLANs associated with each shared port.

Displaying Common Link Counter Information

Each controller and partner node can display counter information for the shared port or ports through which the switch connects to a common link.

- To display *EAPS* shared port counter information, use the following command:

  `show eaps counters shared-port [global | port {segment-port segport {eapsDomain}}]`

  If you specify the global keyword, the switch displays a list of counters that show the totals for all shared ports together. To view the counters for a single shared port, enter the command with the port number.

  If you specify a particular EAPS segment port, the switch displays counter information related to only that segment port for the specified EAPS domain.

Configuration Examples

Migrating from STP to EAPS

This section explains how to migrate or reconfigure an existing *STP* network to an *EAPS* network.

Note
Actual implementation steps on a production network may differ based on the physical topology, switch models, and software versions deployed.

The sample STP network is a simple two-switch topology connected with two Gigabit Ethernet trunk links, which form a broadcast loop. Both Extreme Networks switches are configured for 802.1D mode STP running on a single data *VLAN* named Data. The sample STP network for migration to EAPS is shown in Figure 54.

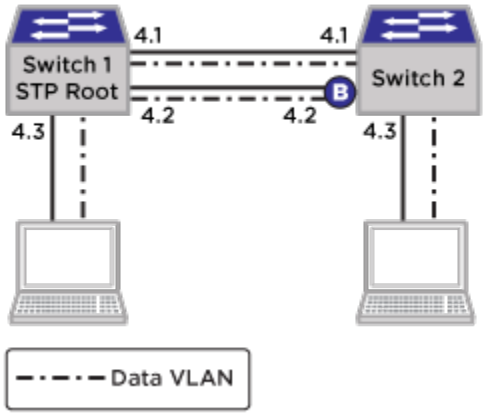


Figure 54: Sample STP Network for Migration to EAPS

Creating and Configuring the EAPS Domain

- Create the *EAPS* domain, configure the switch as a transit node, and define the EAPS primary and secondary ports as shown in the following example:

```
* Edge-Switch#1:1 # create eaps e1-domain
* Edge-Switch#1:2 # configure eaps e1-domain mode transit
* Edge-Switch#1:3 # configure eaps e1-domain primary port 49
* Edge-Switch#1:4 # configure eaps e1-domain secondary port 50
```

Creating and Configuring the EAPS Control VLAN

1. You must create the *EAPS* control *VLAN* and configure the 802.1q tag and ring ports.
2. Configure the control VLANs as part of the EAPS domain. Do this for both switches.

   Switch 1 control VLAN configuration:

```
* SWITCH#1.5 # create vlan control-1
* SWITCH#1.6 # configure vlan control-1 tag 4001
* SWITCH#1.8 # configure vlan control-1 add port 4:1,4:2 tagged
* SWITCH#1.9 # configure eaps new-eaps add control vlan control-1
```

   Switch 2 control VLAN configuration:

```
* SWITCH#2.5 # create vlan control-1
* SWITCH#2.6 # configure vlan control-1 tag 4001
* SWITCH#2.8 # configure vlan control-1 add port 4:1,4:2 tagged
* SWITCH#2.9 # configure eaps new-eaps add control vlan control-1
```

Enabling EAPS and Verify EAPS Status

1. Enable the *EAPS* protocol and the EAPS domain.
2. Confirm that the master node is in Complete state and its secondary port is blocking.

   Switch 1 commands to enable EAPS and the domain:

```
* SWITCH#1.10 # enable eaps
* SWITCH#1.11 # enable eaps new-eaps
```

Switch 2 commands to enable EAPS and verify status:

```
* SWITCH#2.10 # enable eaps
* SWITCH#2.11 # enable eaps new-eaps
* SWITCH#2.12 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: Off
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 1
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain         State           Mo  En  Pri   Sec   Control-Vlan VID   Count
--------------------------------------------------------------------------------
new-eaps       Complete        M   Y   4:1   4:2   control-1    (4001) 0
--------------------------------------------------------------------------------
```

Configuring the STP Protected VLAN as an EAPS Protected VLAN

Configure the data *VLAN* (currently protected by *STP*as an untagged VLAN) as an *EAPS* protected VLAN.

1. Assign an 802.1q tag to the data VLAN, as this might not be required with the previous STP configuration.
2. Next, the data VLAN is added to the EAPS domain as a protected VLAN.
3. Configure the VLAN port changes at the end to prevent any broadcast loop from forming during the transition from STP to EAPS protection.

   A warning message is displayed on the CLI, but this can be ignored, as it is just a reminder that the ring ports have not been added to the protected VLAN yet.
4. Change the port membership for the data VLAN from untagged to 802.1q tagged trunk ports.

   Switch#2 commands to add EAPS protected VLAN and tagged ports:

```
* SWITCH#2.13 # configure vlan data tag 1000
* SWITCH#2.14 # configure new-eaps add protect vlan data
WARNING: Primary port [4:1] is not tagged on vlan "data", EAPS="new-eaps"
WARNING: Secondary port [4:2] is not tagged on vlan "data", EAPS="new-eaps"
* SWITCH#2.15 # configure data add port 4:1,4:2 tagged
```

   Switch#1 commands to add EAPS protected VLAN and tagged ports:

```
* SWITCH#1.13 # configure vlan data tag 1000
* SWITCH#1.14 # configure new-eaps add protect vlan data
WARNING: Primary port [4:1] is not tagged on vlan "data", EAPS="new-eaps"
WARNING: Secondary port [4:2] is not tagged on vlan "data", EAPS="new-eaps"
* SWITCH#1.15 # configure data add port 4:1,4:2 tagged
```

Verifying the EAPS Blocking State for the Protected VLAN

- To ensure there is no potential for a broadcast storm, confirm that *EAPS* is successfully blocking the protected *VLAN*, as shown in the following example:

```
* SWITCH#2.16 # show new-eaps
Name: new-eaps
State: Complete                                Running: Yes
Enabled: Yes Mode: Master
Primary port:   4:1           Port status: Up         Tag status: Tagged
```

```
Secondary port: 4:2           Port status: Blocked    Tag status: Tagged
Hello timer interval: 1  sec  0  millisec
Fail timer interval:  3  sec
Fail Timer expiry action: Send alert
Last valid EAPS update: From Master Id 00:04:96:10:51:50, at Fri Sep 10 13:38:39 2004
EAPS Domain's Controller Vlan: control-1 4001
EAPS Domain's Protected Vlan(s): data 1000
Number of Protected Vlans: 1
```

After you verify that EAPS is protecting the data VLAN, you can safely remove the *STP* configuration.

Verifying the STP Status and Disabling STP

Once you have successfully verified that *EAPS* has taken over loop prevention for the data *VLAN*, you no longer need the *STP* configuration.

Now, verify whether the data VLAN is removed from the STP domain, and then disable the STP protocol.

Switch 2 commands to verify STP status and disable STP:

```
* SWITCH#2.17 # show stp s0
Stpd: s0                Stp: ENABLED            Number of Ports: 0
Rapid Root Failover: Disabled
Operational Mode: 802.1D                        Default Binding Mode: 802.1D
802.1Q Tag: (none)
Ports: (none)
Participating Vlans: (none)
Auto-bind Vlans: Default
Bridge Priority: 32768
BridgeID:               80:00:00:04:96:10:51:50
Designated root:        80:00:00:04:96:10:51:50
RootPathCost: 0         Root Port: ----
MaxAge: 20s             HelloTime: 2s           ForwardDelay: 15s
CfgBrMaxAge: 20s        CfgBrHelloTime: 2s      CfgBrForwardDelay: 15s
Topology Change Time: 35s                       Hold time: 1s
Topology Change Detected: FALSE                 Topology Change: FALSE
Number of Topology Changes: 4
Time Since Last Topology Change: 1435s
* SWITCH#2.18 # show s0 port
Port Mode State Cost Flags Priority Port ID Designated Bridge
* SWITCH#2.19 # disable stp
```

Switch 1 commands to verify STP status and disable STP:

```
* SWITCH#1.16 # show stp s0
Stpd: s0                Stp: ENABLED            Number of Ports: 0
Rapid Root Failover: Disabled
Operational Mode: 802.1D                        Default Binding Mode: 802.1D
802.1Q Tag: (none)
Ports: (none)
Participating Vlans: (none)
Auto-bind Vlans: Default
Bridge Priority: 1
BridgeID:               00:01:00:04:96:10:30:10
Designated root:        00:01:00:04:96:10:30:10
RootPathCost: 0         Root Port: ----
MaxAge: 20s             HelloTime: 2s           ForwardDelay: 15s
CfgBrMaxAge: 20s        CfgBrHelloTime: 2s      CfgBrForwardDelay: 15s
Topology Change Time: 35s                       Hold time: 1s
Topology Change Detected: FALSE                 Topology Change: FALSE
Number of Topology Changes: 2
```

```
Time Since Last Topology Change: 11267s
* SWITCH#1.17 # show stp s0 po
Port Mode State Cost Flags Priority Port ID Designated Bridge
* SWITCH#1.18 # disable stp s0
* SWITCH#1.19 # disable stp
```

The network should now be successfully migrated from STP to EAPS.

Designing and Implementing a Highly Resilient Enterprise Network Using EAPS

Network managers can design and employ a highly resilient end-to-end enterprise network using the Extreme Networks switching platform and the *EAPS* protocol as shown in the following figure.

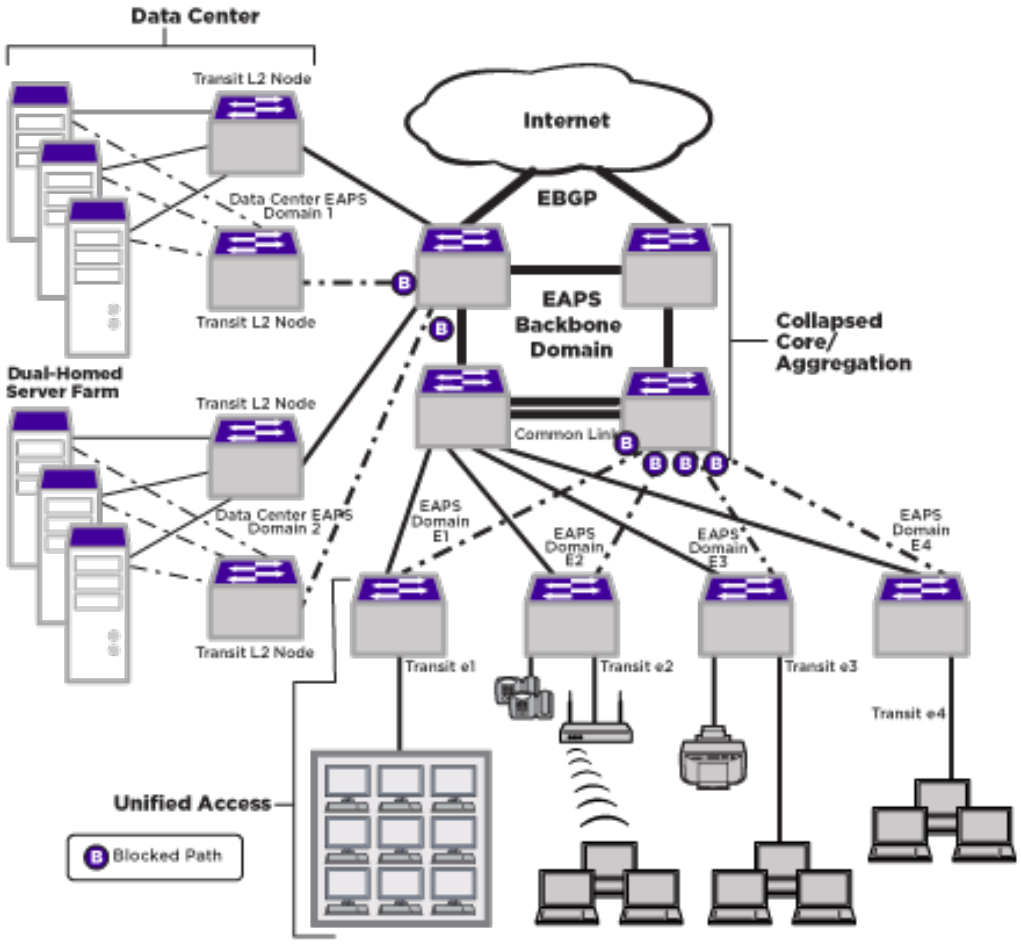


Figure 55: Extreme Networks EAPS Everywhere

EAPS can be used in the network edge to provide link resiliency for Ethernet and IP services in a partial-meshed design. In the aggregation layer, EAPS interconnects multiple edge and core domains. When combined with *VRRP (Virtual Router Redundancy Protocol)* and *OSPF* in the aggregation layer, EAPS provides the foundation for highly resilient IP routing by protecting against link and switch failures.

In the network core, EAPS is used with OSPF to provide a high-performance IP routing backbone with zero downtime or route flaps. Using EAPS and dual-homed server farms in the data center provides high availability for mission-critical server resources.

The collapsed core/aggregation layer and data center also make use of EAPS resilient ring topology to ensure network availability to all critical sources.

Designing and Configuring the Unified Access Layer

The unified access network layer makes use of *EAPS* in a partial-meshed ring topology for maximum resiliency. The edge of the network is the first point of entry for client devices such as PCs, servers, VoIP phones, wireless devices, and printers.

Utilizing EAPS and redundant uplink ports on edge switches increases network resiliency and availability. Edge switches connect their primary and secondary uplink trunk ports to one or more switches in the aggregation network layer (as shown in Figure 56). If the primary uplink port fails, traffic can use the alternate secondary uplink.

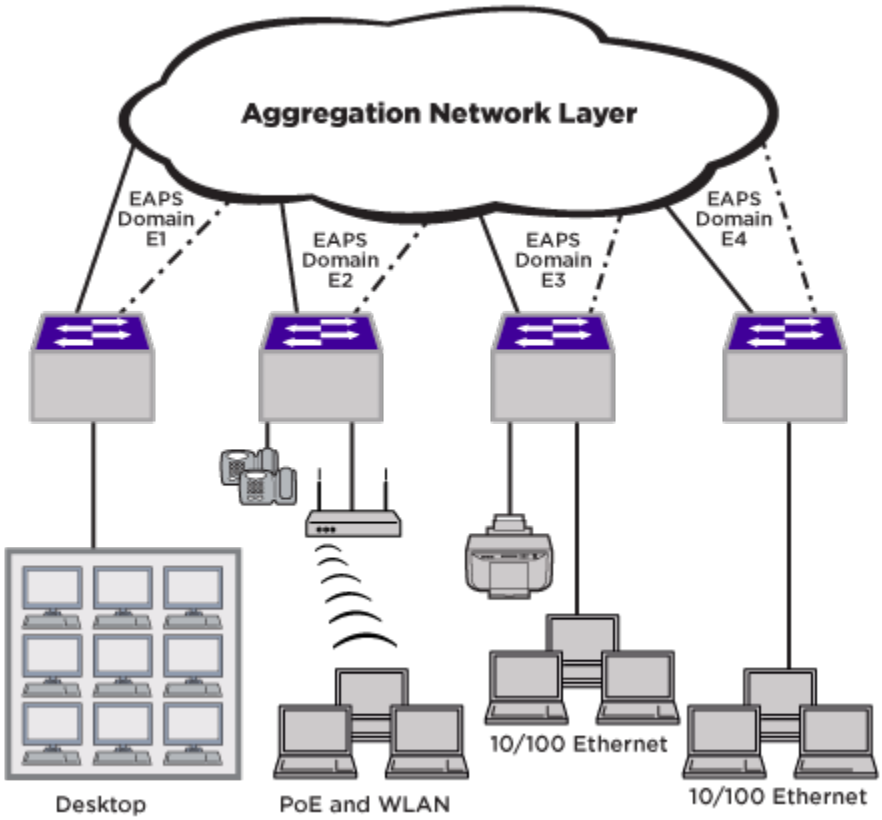


Figure 56: Converged Network Edge (Unified Access Layer)

In this sample network, each edge switch is configured with a unique EAPS domain and control *VLAN*. Protected VLANs can overlap across multiple EAPS domains, or remain local to their own domain.

By putting each edge switch and VLAN into a separate EAPS domain, you gain resiliency and management benefits. First, any link or switch failures in one ring do

not affect the other edge switches. Also, this type of modular design allows you to add edge switches easily without impacting other parts of the network. Troubleshooting becomes easier as the scope of failures can be quickly isolated to a specific EAPS ring or switch.

This section describes how to design the access edge network switches as EAPS transit nodes to provide Ethernet L2 connectivity services. In this example, upstream aggregation switches perform Layer 3 (L3) inter-VLAN routing functions. Although not discussed in the scope of this section, the edge switches could also be configured with additional routing, *QoS*, *WLAN (Wireless Local Area Network)*, or security features.

Creating and Configuring the EAPS Domain

- Create the *EAPS* domain, configure the switch as a transit node, and define the EAPS primary and secondary ports as shown in the following example:

```
* Edge-Switch#1:1 # create eaps e1-domain
* Edge-Switch#1:2 # configure eaps e1-domain mode transit
* Edge-Switch#1:3 # configure eaps e1-domain primary port 49
* Edge-Switch#1:4 # configure eaps e1-domain secondary port 50
```

Creating and Configuring the EAPS Control VLAN

1. You must create the *EAPS* control *VLAN* and configure the 802.1q tag and ring ports.
2. Configure the control VLANs as part of the EAPS domain. Do this for both switches.

   Switch 1 control VLAN configuration:

```
* SWITCH#1.5 # create vlan control-1
* SWITCH#1.6 # configure vlan control-1 tag 4001
* SWITCH#1.8 # configure vlan control-1 add port 4:1,4:2 tagged
* SWITCH#1.9 # configure eaps new-eaps add control vlan control-1
```

   Switch 2 control VLAN configuration:

```
* SWITCH#2.5 # create vlan control-1
* SWITCH#2.6 # configure vlan control-1 tag 4001
* SWITCH#2.8 # configure vlan control-1 add port 4:1,4:2 tagged
* SWITCH#2.9 # configure eaps new-eaps add control vlan control-1
```

Creating and Configuring EAPS Protected VLANs

1. Create at least one *EAPS* protected *VLAN*, and configure its 802.1q tag and ports.
2. Configure the protected *VLAN* as part of the EAPS domain.

   The Protect VLAN contains the EAPS primary and secondary ports as tagged VLAN ports. Additional VLAN ports connected to client devices such as a PC could be untagged or tagged. The following commands accomplish these tasks and should be repeated for all additional protected VLANs:

```
* Edge-Switch#1:10 # create vlan purple-1
* Edge-Switch#1:11 # configure purple-1 tag 1
* Edge-Switch#1:12 # configure purple-1 add port 49,50 tagged
* Edge-Switch#1:13 # configure purple-1 add port 1 untagged
* Edge-Switch#1:14 # configure eaps e1-domain add protect vlan purple-1
```

Enabling the EAPS Protocol and EAPS Domain

- Enable *EAPS* to run on the domain as shown in the following example:

```
* Edge-Switch#1:15 # enable eaps
* Edge-Switch#1:16 # enable eaps e1-domain
```

Verifying the EAPS Configuration and Status

- The command in the following example allows you to verify that the *EAPS* configuration is correct and that the EAPS state is Links-Up.

  Both ring ports must be plugged in to see the Links-Up state.

```
* Edge-Switch#1:17 # show eaps e1-domain detail
Name: "e1-domain" (instance=0) Priority: High
State: Links-Up                Running: Yes
Enabled: Yes                   Mode: Transit
Primary port:   49             Port status: Up         Tag status: Tagged
Secondary port: 50             Port status: Up         Tag status: Tagged
Hello Timer interval: 1 sec  0  millisec
Fail Timer interval: 3 sec
Preforwarding Timer interval: 0 sec
Last valid EAPS update: From Master Id 00:04:96:10:51:50, at Sun Sep 5 23:20:10 2004
EAPS Domain has following Controller Vlan:
Vlan Name                    VID
"control-1"                  4000
EAPS Domain has following Protected Vlan(s):
Vlan Name                    VID
"purple-1"                   0001
Number of Protected Vlans: 1
```

Designing and Configuring the Aggregation Layer

The network switches in the aggregation layer provide additional resiliency benefits.

In the following example, aggregation switches are typically deployed in pairs that protect against single switch failures. Each aggregation switch is physically connected to all edge switches and participates in multiple *EAPS* domains. The aggregation switches can serve a different role within each EAPS domain, with one switch acting as a transit node and the other as a master node.

In this example, we have a common link with overlapping domains (and protected VLANs), which includes an EAPS controller and partner configuration. The result is a partial-mesh network design of EAPS from the access edge to the aggregation layer (see Figure 57).

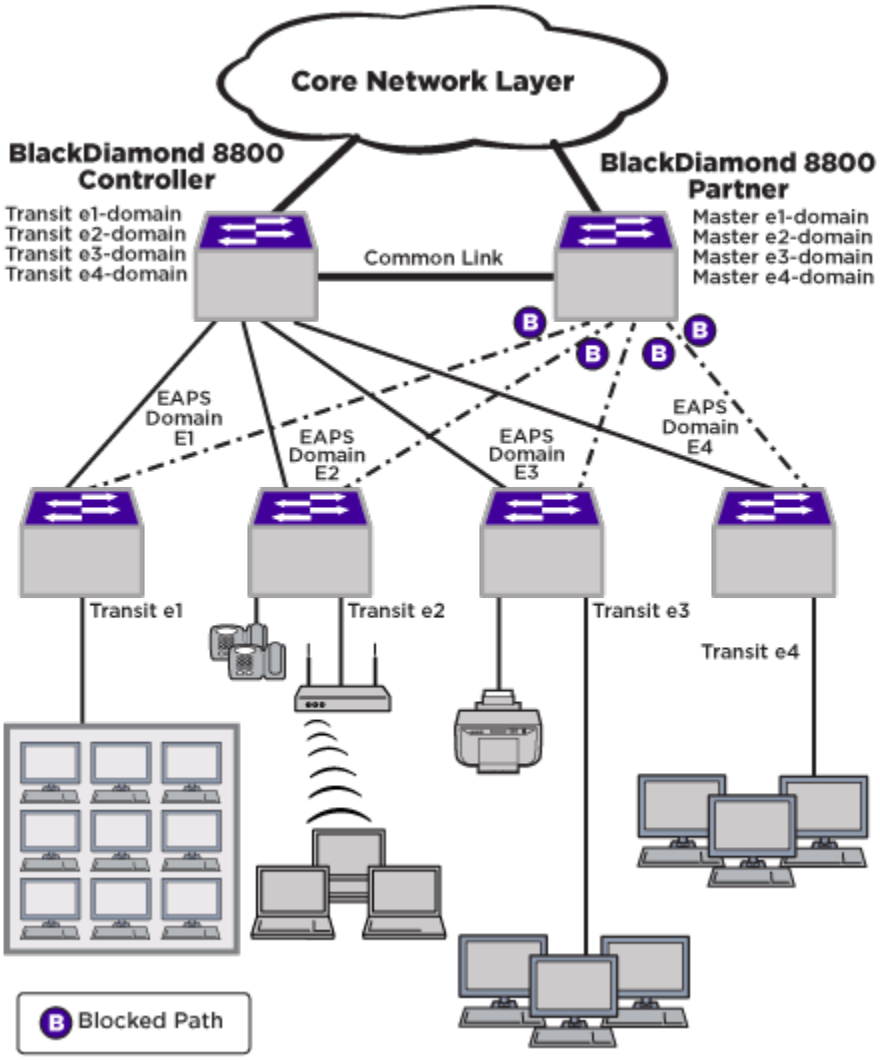


Figure 57: L2 Aggregation Network Layer

The aggregation switches are configured to act as multi-function EAPS nodes to provide L2 connectivity services. After EAPS and L2 connectivity is configured, additional L3 routing configuration can be added.

Using redundant aggregation switches helps protect against a single point of failure at the switch level, while EAPS domains provide fault isolation and minimize the impact that failures have on the network. With shared port configurations, the partial-mesh physical design is maintained without broadcast loops, regardless of where a failure might occur.

To configure the L2 aggregate switches, complete the tasks described in the following sections on all aggregate switches:

1. Create and configure the EAPS domains.
2. Create and configure the EAPS control VLANs.
3. Create and configure the EAPS shared ports.

4. Enable the EAPS protocol and EAPS domain.
5. Create and configure the EAPS protected VLANs.
6. Verify the EAPS configuration and operating state.

Creating and Configuring the EAPS Domains

- Create the *EAPS* domains for each ring (one domain for one edge switch) and configure the EAPS mode.

  Define the primary and secondary ports for each domain. In this example, however, the primary port is the same as the common link. One aggregation switch has EAPS mode configured as master and partner, while the other aggregation switch is configured as transit and controller.

  EAPS master node configuration:

```
* AGG-SWITCH#2.1 # create eaps e1-domain
* AGG-SWITCH#2.2 # create eaps e2-domain
* AGG-SWITCH#2.3 # create eaps e3-domain
* AGG-SWITCH#2.4 # create eaps e4-domain
* AGG-SWITCH#2.5 # configure eaps e1-domain mode master
* AGG-SWITCH#2.6 # configure eaps e2-domain mode master
* AGG-SWITCH#2.7 # configure eaps e3-domain mode master
* AGG-SWITCH#2.8 # configure eaps e4-domain mode master
* AGG-SWITCH#2.9 # configure eaps e1-domain primary port 2:1
* AGG-SWITCH#2.10 # configure eaps e1-domain secondary port 1:1
* AGG-SWITCH#2.11 # configure eaps e2-domain primary port 2:1
* AGG-SWITCH#2.12 # configure eaps e2-domain secondary port 1:4
* AGG-SWITCH#2.13 # configure eaps e3-domain primary port 2:1
* AGG-SWITCH#2.14 # configure eaps e3-domain secondary port 3:1
* AGG-SWITCH#2.15 # configure eaps e4-domain primary port 2:1
* AGG-SWITCH#2.16 # configure eaps e4-domain secondary port 3:2
```

  EAPS transit node configuration:

```
* AGG-SWITCH#1.1 # create eaps e1-domain
* AGG-SWITCH#1.2 # create eaps e2-domain
* AGG-SWITCH#1.3 # create eaps e3-domain
* AGG-SWITCH#1.4 # create eaps e4-domain
* AGG-SWITCH#1.5 # configure eaps e1-domain mode transit
* AGG-SWITCH#1.6 # configure eaps e2-domain mode transit
* AGG-SWITCH#1.7 # configure eaps e3-domain mode transit
* AGG-SWITCH#1.8 # configure eaps e4-domain mode transit
* AGG-SWITCH#1.9 # configure eaps e1-domain primary port 2:1
* AGG-SWITCH#1.10 # configure eaps e1-domain secondary port 1:1
* AGG-SWITCH#1.11 # configure eaps e2-domain primary port 2:1
* AGG-SWITCH#1.12 # configure eaps e2-domain secondary port 1:4
* AGG-SWITCH#1.13 # configure eaps e3-domain primary port 2:1
* AGG-SWITCH#1.14 # configure eaps e3-domain secondary port 3:1
* AGG-SWITCH#1.15 # configure eaps e4-domain primary port 2:1
* AGG-SWITCH#1.16 # configure eaps e4-domain secondary port 3:2
```

Creating and Configuring the EAPS Control VLANs

1. Create the *EAPS* control *VLAN*s (one for each domain) and configure the 802.1q tag and ring ports for each.

2. Configure the control VLANs as part of their respective EAPS domain.

   The control VLAN only contains the EAPS primary and secondary ports configured earlier. The following commands are entered on both aggregate switches:

```
* AGG-SWITCH.17 # create vlan control-1
* AGG-SWITCH.18 # create vlan control-2
* AGG-SWITCH.19 # create vlan control-3
* AGG-SWITCH.20 # create vlan control-4
* AGG-SWITCH.21 # configure vlan control-1 tag 4001
* AGG-SWITCH.22 # configure vlan control-2 tag 4002
* AGG-SWITCH.23 # configure vlan control-3 tag 4003
* AGG-SWITCH.24 # configure vlan control-4 tag 4004
* AGG-SWITCH.29 # configure vlan control-1 add port 2:1,1:1 tagged
* AGG-SWITCH.30 # configure vlan control-2 add port 2:1,1:4 tagged
* AGG-SWITCH.31 # configure vlan control-3 add port 2:1,3:1 tagged
* AGG-SWITCH.32 # configure vlan control-4 add port 2:1,3:2 tagged
* AGG-SWITCH.33 # configure eaps e1-domain add control vlan control-1
* AGG-SWITCH.34 # configure eaps e2-domain add control vlan control-2
* AGG-SWITCH.35 # configure eaps e3-domain add control vlan control-3
* AGG-SWITCH.36 # configure eaps e4-domain add control vlan control-4
```

Creating and Configuring the EAPS Shared Ports

- Create the *EAPS* shared ports, which are used to connect a common-link between the aggregate switches.

  On the first switch, define the shared port mode as partner, and define the link ID. Repeat this step on the other aggregate switch, but configure the shared port mode as controller. The link ID matches the value configured for the partner.

  The following shows an example configuration for the partner:

```
* AGG-SWITCH#2.37 # create eaps shared-port 2:1
* AGG-SWITCH#2.38 # configure eaps shared-port 2:1 mode partner
* AGG-SWITCH#2.39 # configure eaps shared-port 2:1 link-id 21
```

Enabling the EAPS Protocol and EAPS Domain

- Enable *EAPS* to run on the domain as shown in the following example:

```
* Edge-Switch#1:15 # enable eaps
* Edge-Switch#1:16 # enable eaps e1-domain
```

Creating and Configuring the EAPS Protected VLANs

1. Create the *EAPS* protected *VLAN*s for each domain.
2. Configure an 802.1q tag and the ports for each protected VLAN.
3. Configure each protected VLAN as part of the EAPS domain.

   Depending on the scope of the VLAN, it could be added to multiple EAPS domains. This type of VLAN is referred to as an *overlapping protected VLAN*, and requires shared port configurations.

   In this example, there is one overlapping protected VLAN, purple-1, while all other VLANs are isolated to a single EAPS domain (VLANs green-1, orange-1, and red-1). Protected VLAN configuration, such as 802.1q tagging, must match on the edge switch. The commands in the following example are entered on both aggregate switches.

This procedure can also be repeated for additional protected VLANs as needed:

```
* AGG-SWITCH.44 # create vlan purple-1
* AGG-SWITCH.45 # create vlan green-1
* AGG-SWITCH.46 # create vlan orange-1
* AGG-SWITCH.47 # create vlan red-1
* AGG-SWITCH.48 # configure purple-1 tag 1
* AGG-SWITCH.49 # configure green-1 tag 2
* AGG-SWITCH.50 # configure orange-1 tag 3
* AGG-SWITCH.51 # configure red-1 tag 4
* AGG-SWITCH.52 # configure eaps e1-domain add protect vlan purple-1
* AGG-SWITCH.53 # configure eaps e2-domain add protect vlan purple-1
* AGG-SWITCH.54 # configure eaps e3-domain add protect vlan purple-1
* AGG-SWITCH.55 # configure eaps e4-domain add protect vlan purple-1
* AGG-SWITCH.56 # configure eaps e2-domain add protect vlan green-1
* AGG-SWITCH.57 # configure eaps e3-domain add protect vlan orange-1
* AGG-SWITCH.58 # configure eaps e4-domain add protect vlan red-1
* AGG-SWITCH.59 # configure vlan purple-1 add port 2:1,1:1,1:4,3:1,3:2 tagged
* AGG-SWITCH.60 # configure vlan green-1 add port 2:1,1:4 tagged
* AGG-SWITCH.61 # configure vlan orange-1 add port 2:1,3:1 tagged
* AGG-SWITCH.62 # configure vlan red-1 add port 2:1,3:2 tagged
```

Verifying the EAPS Configuration and Operating State

1. When the configuration is complete, confirm that the *EAPS* domain and shared port configuration is correct.
2. Verify whether the EAPS state is Complete and the shared port status is Ready. Both ring ports must be plugged in to see the Links-Up state. This verification is performed on both aggregate switches.

   EAPS master and partner node status verification example:

```
* AGG-SWITCH#2.63 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: Off
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 4
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e1-domain Complete M Y 2:1 1:1 control-1 (4001) 1
e2-domain Complete M Y 2:1 1:4 control-2 (4002) 2
e3-domain Complete M Y 2:1 3:1 control-3 (4003) 2
e4-domain Complete M Y 2:1 3:2 control-4 (4004) 2
--------------------------------------------------------------------------------
* AGG-SWITCH#2.64 # show eaps shared-port
EAPS shared-port count: 1
--------------------------------------------------------------------------------
Link Domain Vlan RB RB
Shared-port Mode Id Up State count count Nbr State Id
--------------------------------------------------------------------------------
2:1 Partner 21 Y Ready 4 4 Yes None None
--------------------------------------------------------------------------------
```

   EAPS transit and controller node status verification example:

```
* AGG-SWITCH#1.63 # show eaps
EAPS Enabled: Yes
```

```
EAPS Fast-Convergence: Off
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 4
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e1-domain Links-Up M Y 2:1 1:1 control-1 (4001) 1
e2-domain Links-Up M Y 2:1 1:4 control-2 (4002) 2
e3-domain Links-Up M Y 2:1 3:1 control-3 (4003) 2
e4-domain Links-Up M Y 2:1 3:2 control-4 (4004) 2
--------------------------------------------------------------------------------
* AGG-SWITCH#1.64 # show eaps shared-port
EAPS shared-port count: 1
--------------------------------------------------------------------------------
Link Domain Vlan RB RB
Shared-port Mode Id Up State count count Nbr State Id
--------------------------------------------------------------------------------
2:1 Controller 21 Y Ready 4 4 Yes None None
--------------------------------------------------------------------------------
```

Designing and Configuring L3 Services on EAPS

This section explains how to run L3 routing services on top of *EAPS* as a foundation.

In this example, *OSPF* is used as the dynamic IP routing protocol to communicate between different *VLAN*s. To provide redundancy at the router level, *VRRP* is used to protect against an aggregation switch failure. VRRP allows one aggregation switch to route IP traffic, and if it fails the other aggregation switch takes over the IP routing role. Each EAPS protected VLAN provides L3 connectivity to the clients by configuring IP addressing, OSPF routing, and VRRP on the aggregation switches.

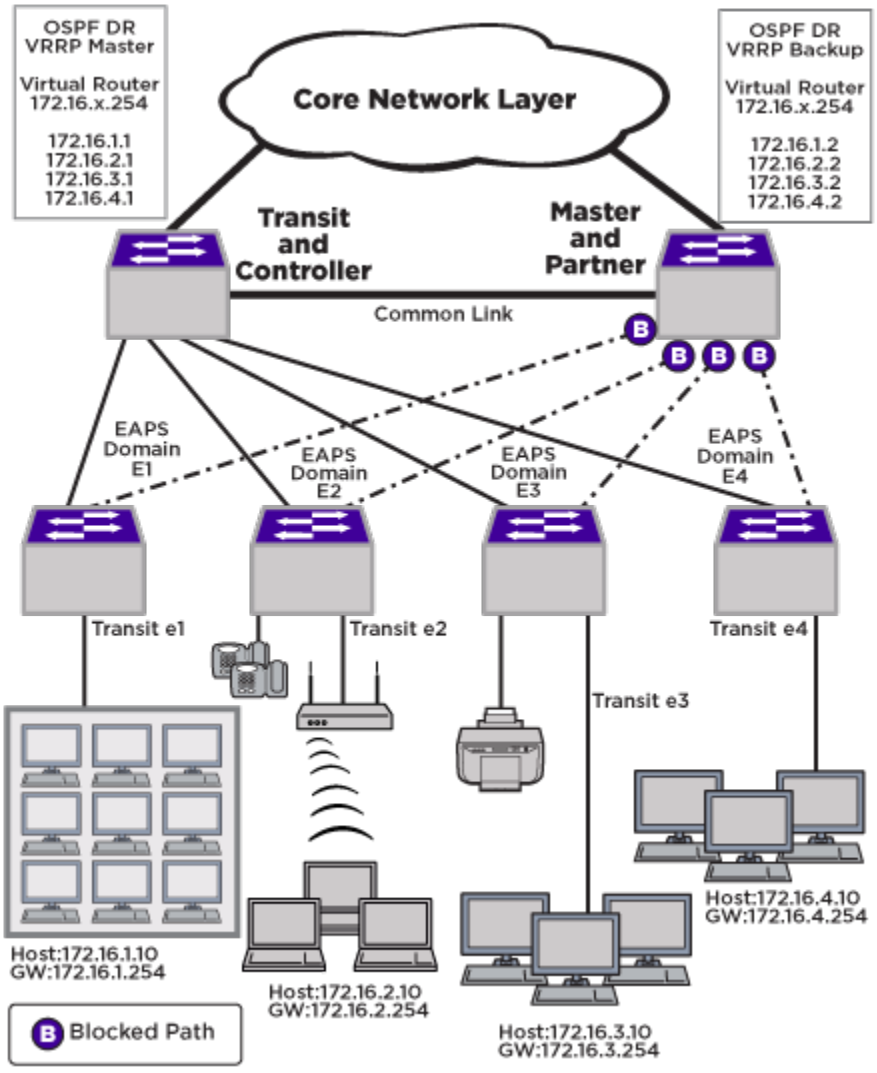


Figure 58: L2 and L3 Aggregation Network Layer

IP routing is added to the design on the access network switches by configuring each EAPS protected VLAN as an OSPF interface. Because these are broadcast OSPF interfaces, we need to specify a Designated Router (DR) and Backup Designated Router (BDR). While the EAPS transit and controller node is not blocking any ports, it is configured as the OSPF DR.

The EAPS master and partner node is then configured as the BDR. Similarly, the EAPS transit and controller node is also configured as the VRRP master, which provides L3 routing to the hosts. The EAPS master and partner node is configured as the VRRP backup router for redundancy.

Using redundant aggregation switches with VRRP protects against a single point of failure at the switch level. Client devices receive non-stop IP routing services in the event of link or aggregation switch failure without any reconfiguration. OSPF provides fast convergence from any routing failures. EAPS provides the resilient L2 foundation and minimizes the occurrence of routing interface flaps or dropped OSPF neighbor adjacencies.

To configure L3 on the aggregation switches, completed the tasks described in the following sections:

1. Configure OSPF on the EAPS protected VLANs.
2. Configure OSPF on the EAPS protected VLANs.
3. Configure VRRP on the EAPS protected VLANs.
4. Verify OSPF and VRRP configuration status.

Configuring IP Addresses on the EAPS Protected VLANs

Client host stations need the IP address configuration to match their protected VLANs. The edge switches do not require IP addresses, but this could optionally be done for management or troubleshooting purposes.

The following example shows IP address configuration:

```
* AGG-SWITCH#1.1 # configure vlan green-1 ipaddress 172.16.1.1/24
* AGG-SWITCH#1.2 # configure vlan purple-1 ipaddress 172.16.2.1/24
* AGG-SWITCH#1.3 # configure vlan orange-1 ipaddress 172.16.3.1/24
* AGG-SWITCH#1.4 # configure vlan red-1 ipaddress 172.16.4.1/24
* AGG-SWITCH#2.1 # configure vlan green-1 ipaddress 172.16.1.2/24
* AGG-SWITCH#2.2 # configure vlan purple-1 ipaddress 172.16.2.2/24
* AGG-SWITCH#2.3 # configure vlan orange-1 ipaddress 172.16.3.2/24
* AGG-SWITCH#2.4 # configure vlan red-1 ipaddress 172.16.4.2/24
```

Configuring OSPF on the EAPS Protected VLANs

Because *OSPF* broadcast networks are being used, configure the DR and BDR for each *VLAN*. Configure the *EAPS* transit and controller as the DR by using a higher OSPF priority value since it is not performing L2 blocking. The EAPS master and partner switch is configured as the BDR. In this example, all edge EAPS protected VLANs are placed in the OSPF backbone area, but another OSPF area could be created if desired.

Example OSPF DR configuration:

```
* AGG-SWITCH#1.5 # enable ipforwarding vlan green-1
* AGG-SWITCH#1.6 # enable ipforwarding vlan purple-1
* AGG-SWITCH#1.7 # enable ipforwarding vlan orange-1
* AGG-SWITCH#1.8 # enable ipforwarding vlan red-1
* AGG-SWITCH#1.9 # configure ospf routerid 172.16.1.1
* AGG-SWITCH#1.10 # configure ospf add vlan green-1 area 0.0.0.0
* AGG-SWITCH#1.11 # configure ospf add vlan purple-1 area 0.0.0.0
* AGG-SWITCH#1.12 # configure ospf add vlan orange-1 area 0.0.0.0
* AGG-SWITCH#1.13 # configure ospf add vlan red-1 area 0.0.0.0
* AGG-SWITCH#1.14 # configure ospf vlan green-1 priority 110
* AGG-SWITCH#1.15 # configure ospf vlan purple-1 priority 110
* AGG-SWITCH#1.16 # configure ospf vlan orange-1 priority 110
* AGG-SWITCH#1.17 # configure ospf vlan red-1 priority 110
* AGG-SWITCH#1.18 # enable ospf
```

Example OSPF BDR configuration:

```
* AGG-SWITCH#2.5 # enable ipforwarding vlan green-1
* AGG-SWITCH#2.6 # enable ipforwarding vlan purple-1
* AGG-SWITCH#2.7 # enable ipforwarding vlan orange-1
* AGG-SWITCH#2.8 # enable ipforwarding vlan red-1
* AGG-SWITCH#2.9 # configure ospf routerid 172.16.1.2
* AGG-SWITCH#2.10 # configure ospf add vlan green-1 area 0.0.0.0
* AGG-SWITCH#2.11 # configure ospf add vlan purple-1 area 0.0.0.0
* AGG-SWITCH#2.12 # configure ospf add vlan orange-1 area 0.0.0.0
```

```
* AGG-SWITCH#2.13 # configure ospf add vlan red-1 area 0.0.0.0
* AGG-SWITCH#2.14 # configure ospf vlan green-1 priority 100
* AGG-SWITCH#2.15 # configure ospf vlan purple-1 priority 100
* AGG-SWITCH#2.16 # configure ospf vlan orange-1 priority 100
* AGG-SWITCH#2.17 # configure ospf vlan red-1 priority 100
* AGG-SWITCH#2.18 # enable ospf
```

Configuring VRRP on the EAPS Protected VLANs

The VRRP is configured with the virtual IP address of 172.16.x.254 for each *VLAN* (example VLAN green-1 = 172.16.1.254). The *VRRP* virtual router IP address is configured as the default gateway of each client machine. Since it is not performing L2 blocking, configure the *EAPS* transit and controller as VRRP master router by using a higher priority value. The EAPS master and partner switch is configured as the VRRP backup router.

Example VRRP master router configuration:

```
* AGG-SWITCH#1.19 # create vrrp vlan green-1 vrid 1
* AGG-SWITCH#1.20 # configure vrrp vlan green-1 vrid 1 priority 110
* AGG-SWITCH#1.21 # configure vrrp vlan green-1 vrid 1 add 172.16.1.254
* AGG-SWITCH#1.22 # enable vrrp vlan green-1 vrid 1
* AGG-SWITCH#1.23 # create vrrp vlan purple-1 vrid 1
* AGG-SWITCH#1.24 # configure vrrp vlan purple-1 vrid 1 priority 110
* AGG-SWITCH#1.25 # configure vrrp vlan purple-1 vrid 1 add 172.16.2.254
* AGG-SWITCH#1.26 # enable vrrp vlan purple-1 vrid 1
* AGG-SWITCH#1.27 # create vrrp vlan orange-1 vrid 1
* AGG-SWITCH#1.28 # configure vrrp vlan orange-1 vrid 1 priority 110
* AGG-SWITCH#1.29 # configure vrrp vlan orange-1 vrid 1 add 172.16.3.254
* AGG-SWITCH#1.30 # enable vrrp vlan orange-1 vrid 1
* AGG-SWITCH#1.31 # create vrrp vlan red-1 vrid 1
* AGG-SWITCH#1.32 # configure vrrp vlan red-1 vrid 1 priority 110
* AGG-SWITCH#1.33 # configure vrrp vlan red-1 vrid 1 add 172.16.4.254
* AGG-SWITCH#1.34 # enable vrrp vlan red-1 vrid 1
```

Example VRRP backup router configuration:

```
* AGG-SWITCH#2.19 # create vrrp vlan green-1 vrid 1
* AGG-SWITCH#2.20 # configure vrrp vlan green-1 vrid 1 priority 100
* AGG-SWITCH#2.21 # configure vrrp vlan green-1 vrid 1 add 172.16.1.254
* AGG-SWITCH#2.22 # enable vrrp vlan green-1 vrid 1
* AGG-SWITCH#2.23 # create vrrp vlan purple-1 vrid 1
* AGG-SWITCH#2.24 # configure vrrp vlan purple-1 vrid 1 priority 100
* AGG-SWITCH#2.25 # configure vrrp vlan purple-1 vrid 1 add 172.16.2.254
* AGG-SWITCH#2.26 # enable vrrp vlan purple-1 vrid 1
* AGG-SWITCH#2.27 # create vrrp vlan orange-1 vrid 1
* AGG-SWITCH#2.28 # configure vrrp vlan orange-1 vrid 1 priority 100
* AGG-SWITCH#2.29 # configure vrrp vlan orange-1 vrid 1 add 172.16.3.254
* AGG-SWITCH#2.30 # enable vrrp vlan orange-1 vrid 1
* AGG-SWITCH#2.31 # create vrrp vlan red-1 vrid 1
* AGG-SWITCH#2.32 # configure vrrp vlan red-1 vrid 1 priority 100
* AGG-SWITCH#2.33 # configure vrrp vlan red-1 vrid 1 add 172.16.4.254
* AGG-SWITCH#2.34 # enable vrrp vlan red-1 vrid 1
```

Verifying OSPF and VRRP Configuration Status

1. Verify the *OSPF* neighbor adjacencies are established and that the DR and BDR status is correct.

2. Verify that the *VRRP* is running and the VRRP master/backup status is correct.

   OSPF and VRRP verification example:

```
* AGG-SWITCH#1.35 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
172.16.1.2 100 FULL /BDR 00:18:01:08/00:00:00:03 172.16.3.2 orange-1
172.16.1.2 100 FULL /BDR 00:18:01:08/00:00:00:03 172.16.4.2 red-1
172.16.1.2 100 FULL /BDR 00:17:54:17/00:00:00:03 172.16.1.2 green-1
172.16.1.2 100 FULL /BDR 00:17:54:07/00:00:00:03 172.16.2.2 purple-1
* AGG-SWITCH#1.36 # show vrrp
VLAN Name VRID Pri Virtual IP Addr State Master Mac Address TP/TR/TV/P/T
green-1(En) 0001 110 172.16.1.254 MSTR 00:00:5e:00:01:01 0 0 0 Y 1
purple-(En) 0001 110 172.16.2.254 MSTR 00:00:5e:00:01:01 0 0 0 Y 1
orange-(En) 0001 110 172.16.3.254 MSTR 00:00:5e:00:01:01 0 0 0 Y 1
red-1(En) 0001 110 172.16.4.254 MSTR 00:00:5e:00:01:01 0 0 0 Y 1
En-Enabled, Ds-Disabled, Pri-Priority, T-Advert Timer, P-Preempt
TP-Tracked Pings, TR-Tracked Routes, TV-Tracked VLANs
* AGG-SWITCH#2.35 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
172.16.1.1 110 FULL /DR 00:18:01:08/00:00:00:03 172.16.3.1 orange-1
172.16.1.1 110 FULL /DR 00:18:01:08/00:00:00:03 172.16.4.1 red-1
172.16.1.1 110 FULL /DR 00:17:54:17/00:00:00:03 172.16.1.1 green-1
172.16.1.1 110 FULL /DR 00:17:54:07/00:00:00:03 172.16.2.1 purple-1
* AGG-SWITCH#2.36 # show vrrp
VLAN Name VRID Pri Virtual IP Addr State Master Mac Address TP/TR/TV/P/T
green-1(En) 0001 100 172.16.1.254 BKUP 00:00:5e:00:01:01 0 0 0 Y 1
purple-(En) 0001 100 172.16.2.254 BKUP 00:00:5e:00:01:01 0 0 0 Y 1
orange-(En) 0001 100 172.16.3.254 BKUP 00:00:5e:00:01:01 0 0 0 Y 1
red-1(En) 0001 100 172.16.4.254 BKUP 00:00:5e:00:01:01 0 0 0 Y 1
En-Enabled, Ds-Disabled, Pri-Priority, T-Advert Timer, P-Preempt
TP-Tracked Pings, TR-Tracked Routes, TV-Tracked VLANs
```

Designing and Configuring the Core Layer with EAPS

The core switches provide high performance backbone routing between the edge, aggregation, data center, and external Internet networks.

An additional high availability backbone ring is built that combines *EAPS* and *OSPF*. Using EAPS and OSPF together increases the stability of IP routing tables. Since EAPS provides 50-millisecond convergence for link failures, OSPF adjacencies do not flap. In this example, the backbone ring is formed by adding two core L2/L3 switches and connecting them to the two existing aggregation switches. The core switches also provide routing to the Internet using BGP (see Figure 59).

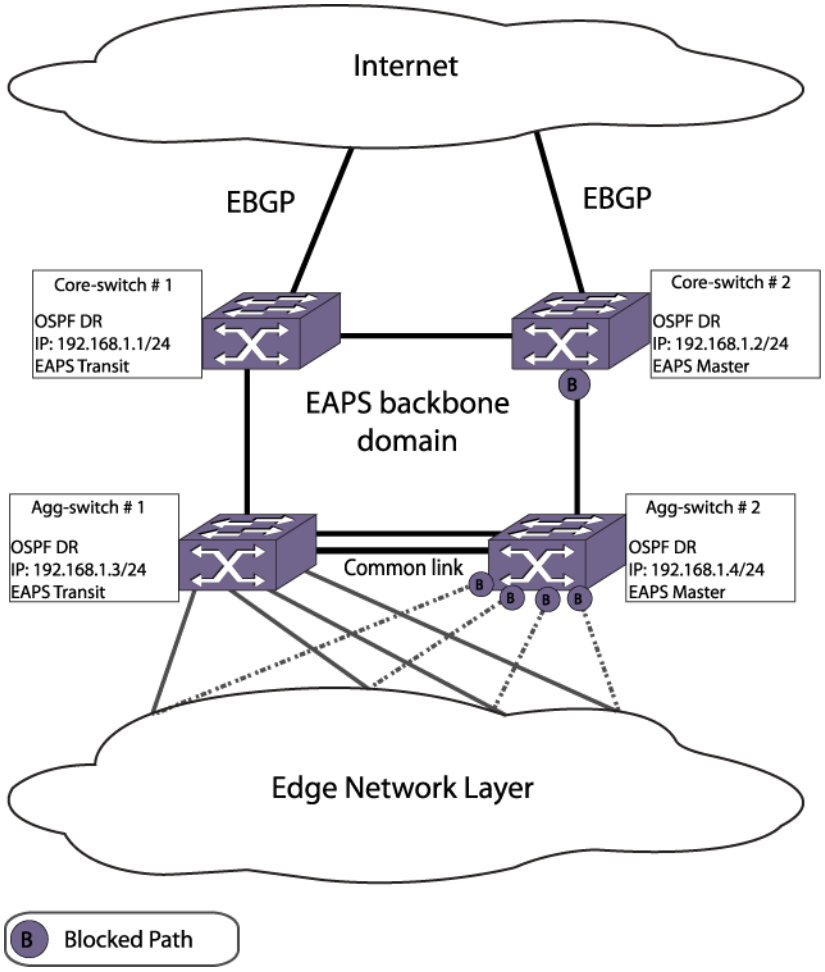


Figure 59: Core EAPS and OSPF Network Layer

Using redundant core switches protects against a single point of failure at the switch level. OSPF provides fast convergence from any routing failures. EAPS provides the resilient L2 foundation and minimizes the occurrence of routing interface flaps or dropped OSPF neighbor adjacencies. Combining EAPS and OSPF provides the highest level of network resiliency and routing stability.

Configuring the core switches requires a new EAPS domain with a single EAPS protected *VLAN* with OSPF forming the backbone IP network. Additional configuration is needed on the aggregation switches to connect them to the backbone EAPS and OSPF ring. Since the steps are similar to previous configuration examples, the L2 (EAPS) and L3 (OSPF) configurations are combined. Since the *BGP* configuration is independent of EAPS configuration, BGP configuration is not discussed here.

To configure backbone connectivity on the core and aggregation switches, complete the tasks described in the following sections:

1. Create and configure the backbone EAPS domain.
2. Create and configure the backbone EAPS protected VLANs.
3. Configure an IP address and OSPF on the backbone VLAN.
4. Verify EAPS and OSPF configuration status.

Creating and Configuring the Backbone EAPS Domain

1. Create the backbone *EAPS* domains and configure the EAPS mode.
2. Define the primary and secondary ports for each domain.
   Configure on both core and aggregation switches.

   Core-Switch 1 EAPS configuration:

```
* CORE-SWITCH#1.1 # create eaps e5-domain
* CORE-SWITCH#1.2 # configure eaps e5-domain mode transit
* CORE-SWITCH#1.3 # configure eaps e5-domain primary port 2:1
* CORE-SWITCH#1.4 # configure eaps e5-domain secondary port 2:4
```

   Core-Switch 2 EAPS configuration:

```
* CORE-SWITCH#2.1 # create eaps e5-domain
* CORE-SWITCH#2.2 # configure eaps e5-domain mode master
* CORE-SWITCH#2.3 # configure eaps e5-domain primary port 2:1
* CORE-SWITCH#2.4 # configure eaps e5-domain secondary port 2:4
```

   Agg-Switch 1 EAPS configuration:

```
* AGG-SWITCH#1.1 # create eaps e5-domain
* AGG-SWITCH#1.2 # configure eaps e5-domain mode transit
* AGG-SWITCH#1.3 # configure eaps e5-domain primary port 2:1
* AGG-SWITCH#1.4 # configure eaps e5-domain secondary port 2:4
```

   Agg-Switch 2 EAPS configuration:

```
* AGG-SWITCH#2.1 # create eaps e5-domain
* AGG-SWITCH#2.2 # configure eaps e5-domain mode transit
* AGG-SWITCH#2.3 # configure eaps e5-domain primary port 2:1
* AGG-SWITCH#2.4 # configure eaps e5-domain secondary port 2:4
```

Creating and Configuring the Backbone EAPS Control VLAN

1. Create the *EAPS* control *VLAN* and configure its 802.1q tag, and ring ports.
2. Configure the control VLANs as part of the backbone EAPS domain. Enable EAPS and the backbone EAPS domain. Configure on both core and aggregation switches (EAPS is already enabled on aggregation switches).
   Core-Switch#1 control VLAN configuration:

```
* CORE-SWITCH#1.1 # create vlan control-5
* CORE-SWITCH#1.2 # configure vlan control-5 tag 4005
* CORE-SWITCH#1.4 # configure vlan control-5 add port 2:1,2:4 tagged
* CORE-SWITCH#1.5 # configure eaps e5-domain add control vlan control-5
* CORE-SWITCH#1.6 # enable eaps
* CORE-SWITCH#1.7 # enable eaps e5-domain
```

   Core-Switch#2 control VLAN configuration:

```
* CORE-SWITCH#2.1 # create vlan control-5
* CORE-SWITCH#2.2 # configure vlan control-5 tag 4005
* CORE-SWITCH#2.4 # configure vlan control-5 add port 2:1,2:4 tagged
* CORE-SWITCH#2.5 # configure eaps e5-domain add control vlan control-5
* CORE-SWITCH#2.6 # enable eaps
* CORE-SWITCH#2.7 # enable eaps e5-domain
```

   Agg-Switch#1 control VLAN configuration:

```
* AGG-SWITCH#1.1 # create vlan control-5
* AGG-SWITCH#1.2 # configure vlan control-5 tag 4005
* AGG-SWITCH#1.4 # configure vlan control-5 add port 2:4,2:6 tagged
* AGG-SWITCH#1.5 # configure eaps e5-domain add control vlan control-5
* AGG-SWITCH#1.6 # enable eaps e5-domain
```

Agg-Switch#2 control VLAN configuration:

```
* AGG-SWITCH#2.1 # create vlan control-5
* AGG-SWITCH#2.2 # configure vlan control-5 tag 4005
* AGG-SWITCH#2.4 # configure vlan control-5 add port 2:4,2:6 tagged
* AGG-SWITCH#2.5 # configure eaps e5-domain add control vlan control-5
* AGG-SWITCH#1.6 # enable eaps e5-domain
```

Creating and Configuring the Backbone EAPS Protected VLANs

1. Create the *EAPS* protected *VLAN* for the backbone domain.
2. Configure the 802.1q tag and ports for the protected VLANs.

   Because this VLAN is only used for transit routing, there are no other ports besides the ring ports.
3. Configure the protected VLAN as part of the EAPS domain. Do this configuration on both the core and aggregate switches.

   Core-Switch#1 protected VLAN configuration:

```
* CORE-SWITCH#1.8 # create vlan backbone
* CORE-SWITCH#1.9 # configure vlan backbone tag 3000
* CORE-SWITCH#1.10 # configure vlan backbone add port 2:1,2:4 tagged
* CORE-SWITCH#1.11 # configure eaps e5-domain add protect vlan backbone
```

   Core-Switch#2 protected VLAN configuration:

```
* CORE-SWITCH#2.8 # create vlan backbone
* CORE-SWITCH#2.9 # configure vlan backbone tag 3000
* CORE-SWITCH#2.10 # configure vlan backbone add port 2:1,2:4 tagged
* CORE-SWITCH#2.11 # configure eaps e5-domain add protect vlan backbone
```

   Agg-Switch#1 protected VLAN configuration:

```
* AGG-SWITCH#1.7 # create vlan backbone
* AGG-SWITCH#1.8 # configure vlan backbone tag 3000
* AGG-SWITCH#1.9 # configure vlan backbone add port 2:4,2:6 tagged
* AGG-SWITCH#1.10 # configure eaps e5-domain add protect vlan backbone
```

   Agg-Switch#2 protected VLAN configuration:

```
* AGG-SWITCH#2.7 # create vlan backbone
* AGG-SWITCH#2.8 # configure vlan backbone tag 3000
* AGG-SWITCH#2.9 # configure vlan backbone add port 2:4,2:6 tagged
* AGG-SWITCH#2.10 # configure eaps e5-domain add protect vlan backbone
```

Configuring an IP Address and OSPF on the Backbone VLAN

1. Configure an IP address and enable IP forwarding (routing) on the backbone protected *VLAN*.
2. *OSPF* is configured and because an OSPF broadcast network is used, configure the designated router and backup designated router for each VLAN.

   Since it is not performing L2 blocking, configure the *EAPS* transit core switch as the DR by using a higher OSPF priority value. The EAPS master core switch is configured as the BDR. The aggregation transit switches need not perform DR/BDR duties for the backbone VLAN, so their OSPF priority is configured at 0 to force ODR behavior.

   Core-Switch#1 OSPF configuration:

```
* CORE-SWITCH#1.12 # configure vlan backbone ipaddress 192.168.1.1/24
* CORE-SWITCH#1.13 # enable ipforwarding vlan backbone
```

```
* CORE-SWITCH#1.14 # configure ospf routerid 192.168.1.1
* CORE-SWITCH#1.15 # configure ospf add vlan backbone area 0.0.0.0
* CORE-SWITCH#1.16 # configure ospf vlan backbone priority 110
* CORE-SWITCH#1.17 # enable ospf
```

Core-Switch#2 OSPF configuration:

```
* CORE-SWITCH#2.12 # configure vlan backbone ipaddress 192.168.1.2/24
* CORE-SWITCH#2.13 # enable ipforwarding vlan backbone
* CORE-SWITCH#2.14 # configure ospf routerid 192.168.1.2
* CORE-SWITCH#2.15 # configure ospf add vlan backbone area 0.0.0.0
* CORE-SWITCH#2.16 # configure ospf vlan backbone priority 100
* CORE-SWITCH#2.17 # enable ospf
```

Agg-Switch#1 OSPF configuration:

```
* AGG-SWITCH#1.11 # configure vlan backbone ipaddress 192.168.1.3/24
* AGG-SWITCH#1.12 # enable ipforwarding vlan backbone
* AGG-SWITCH#1.13 # configure ospf add vlan backbone area 0.0.0.0
* AGG-SWITCH#1.14 # configure ospf vlan backbone priority 0
```

Agg-Switch#2 OSPF configuration:

```
* AGG-SWITCH#2.11 # configure vlan backbone ipaddress 192.168.1.4/24
* AGG-SWITCH#2.12 # enable ipforwarding vlan backbone
* AGG-SWITCH#2.13 # configure ospf add vlan backbone area 0.0.0.0
* AGG-SWITCH#2.14 # configure ospf vlan backbone priority 0
```

Verifying EAPS and OSPF Configuration Status

1. Verify that the backbone *EAPS* domain and *OSPF* configuration is correct.
2. Confirm that the OSPF neighbor adjacencies and DR/BDR/ODR status are correct. Verify this status on both aggregate switches.

   Core-Switch#1 EAPS and OSPF status example:

```
* CORE-SWITCH#1.18 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 1
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e5-domain Links-Up T Y 2:1 2:4 control-5 (4005) 1
--------------------------------------------------------------------------------
* CORE-SWITCH#1.19 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
192.168.1.3 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.3 backbone
192.168.1.4 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.4 backbone
192.168.1.2 100 FULL /BDR 00:05:23:17/00:00:00:09 192.168.1.2 backbone
```

   Core-Switch#2 EAPS and OSPF status example:

```
* CORE-SWITCH#2.18 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
```

```
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 1
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e5-domain Complete T Y 2:1 2:4 control-5 (4005) 1
--------------------------------------------------------------------------------
* CORE-SWITCH#2.19 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
192.168.1.3 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.3 backbone
192.168.1.4 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.4 backbone
192.168.1.1 110 FULL /DR 00:05:23:17/00:00:00:09 192.168.1.1 backbone
```

Agg-Switch#1 EAPS and OSPF status example:

```
* AGG-SWITCH#1.15 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 5
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e1-domain Links-Up T Y 1:1 2:1 control-1 (4001) 2
e2-domain Links-Up T Y 1:4 2:1 control-2 (4002) 2
e3-domain Links-Up T Y 3:1 2:1 control-3 (4003) 2
e4-domain Links-Up T Y 3:2 2:1 control-4 (4004) 2
e5-domain Links-Up T Y 2:4 2:6 control-5 (4005) 1
--------------------------------------------------------------------------------
* AGG-SWITCH#1.16 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
192.168.1.1 110 FULL /DR 00:00:28:51/00:00:00:01 192.168.1.1 backbone
192.168.1.2 100 FULL /BDR 00:00:28:51/00:00:00:01 192.168.1.2 backbone
192.168.1.4 0 2WAY /DROTHER00:05:45:40/00:00:00:03 192.168.1.4 backbone
172.16.1.2 100 FULL /BDR 00:18:01:08/00:00:00:03 172.16.3.2 orange-1
172.16.1.2 100 FULL /BDR 00:18:01:08/00:00:00:03 172.16.4.2 red-1
172.16.1.2 100 FULL /BDR 00:17:54:17/00:00:00:03 172.16.1.2 green-1
172.16.1.2 100 FULL /BDR 00:17:54:07/00:00:00:03 172.16.2.2 purple-1
```

Agg-Switch#2 EAPS and OSPF status example:

```
* AGG-SWITCH#2.15 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 5
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e1-domain Complete M Y 2:1 1:1 control-1 (4001) 2
e2-domain Complete M Y 2:1 1:4 control-2 (4002) 2
```

```
e3-domain Complete M Y 2:1 3:1 control-3 (4003) 2
e4-domain Complete M Y 2:1 3:2 control-4 (4004) 2
e5-domain Links-Up T Y 2:4 2:6 control-5 (4005) 1
------------------------------------------------------------------------------
* AGG-SWITCH#2.16 # show ospf neighbor
Interface
192.168.1.1 110 FULL /DR 00:00:28:51/00:00:00:01 192.168.1.1 backbone
192.168.1.2 100 FULL /BDR 00:00:28:51/00:00:00:01 192.168.1.2 backbone
192.168.1.3 0 2WAY /DROTHER00:05:45:40/00:00:00:03 192.168.1.3 backbone
172.16.1.1 110 FULL /DR 00:18:01:08/00:00:00:03 172.16.3.1 orange-1
172.16.1.1 110 FULL /DR 00:18:01:08/00:00:00:03 172.16.4.1 red-1
172.16.1.1 110 FULL /DR 00:17:54:17/00:00:00:03 172.16.1.1 green-1
172.16.1.1 110 FULL /DR 00:17:54:07/00:00:00:03 172.16.2.1 purple-1
```

Designing and Configuring the Data Center Switches with EAPS

Building from the network core, you can expand the network with additional *EAPS* rings to provide resiliency to mission-critical server farms.

The core switches provide high performance backbone routing between the data center and the rest of the network, which includes both internal and external (Internet) destinations. The core switch acts as the EAPS master node for each ring, while the data center switches act as EAPS transit nodes to complete the ring. The core switch also acts as the *OSPF* routing node to provide gateway routing functionality to the server-farms. For an additional level of resiliency, each server is dual-homed (dual attached) to both EAPS transit L2 switches. Even if a switch or link fails, the servers are available.

The network design and configuration is similar to the edge and aggregation EAPS and OSPF layers. The modular approach is simple and scalable, and allows additional data center rings to be added to provide room for growth. In our example, server-farms are isolated into separate categories such as external and internal service groups, which yield additional security and resiliency benefits.

To configure the data center switches, you need a new EAPS domain with a single EAPS protected *VLAN* to form the server-farm network. In this example, two data center switches are configured as EAPS transit nodes (L2 switch only) and attach to the existing core switch acting as the EAPS master. Each server in the server-farm is dual-homed to both EAPS transit switches in the data center for additional physical resiliency. IP routing functionality is performed by the core switch via OSPF, which provides L3 connectivity to the rest of the network.

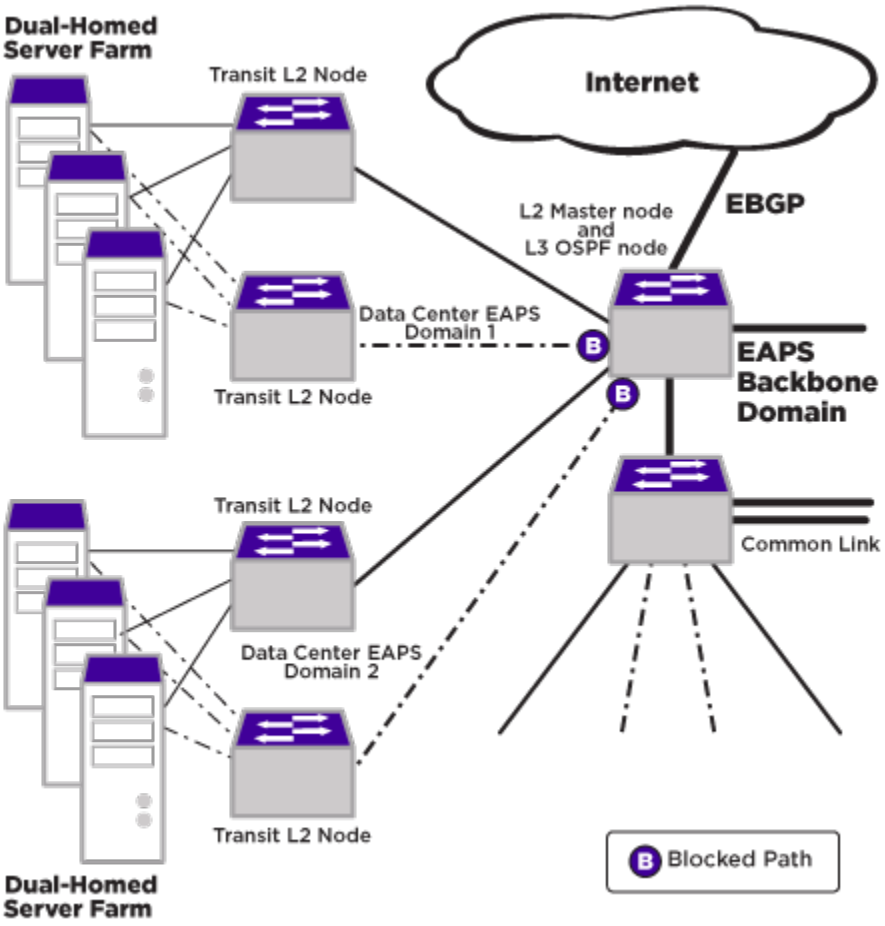


Figure 60: Data Center EAPS and OSPF Network Layer

To configure data center connectivity, complete the tasks described in the following sections:

1. Create and configure the data center EAPS domain.
2. Create and configure the data center EAPS Control VLAN.
3. Create and configure the data center EAPS protected VLANs.
4. Configure an IP address and OSPF on the backbone VLAN.
5. Verify EAPS and OSPF configuration status.

Creating and Configuring the Data Center EAPS Domain

Create the backbone *EAPS* domains, configure the EAPS mode, and define the primary and secondary ports for each domain. Do this configuration on both core and aggregation switches.

Core-Switch#1 EAPS configuration:

```
* CORE-SWITCH#1.1 # create eaps e6-domain
* CORE-SWITCH#1.2 # configure eaps e6-domain mode master
* CORE-SWITCH#1.3 # configure eaps e6-domain primary port 4:1
* CORE-SWITCH#1.4 # configure eaps e6-domain secondary port 4:2
```

Data center-Switch#1 EAPS configuration:

```
* DC-SWITCH#1.1 # create eaps e6-domain
* DC-SWITCH#1.2 # configure eaps e6-domain mode transit
* DC-SWITCH#1.3 # configure eaps e6-domain primary port 49
* DC-SWITCH#1.4 # configure eaps e6-domain secondary port 50
```

Datacenter -Switch#2 EAPS configuration:

```
* DC-SWITCH#2.1 # create eaps e6-domain
* DC-SWITCH#2.2 # configure eaps e6-domain mode transit
* DC-SWITCH#2.3 # configure eaps e6-domain primary port 49
* DC-SWITCH#2.4 # configure eaps e6-domain secondary port 50
```

Creating and Configuring the Data Center EAPS Control VLAN

1. Create the *EAPS* control *VLAN* and configure its 802.1q tag, and ring ports.
2. Configure the control VLANs as part of the data center EAPS domain. Enable EAPS and the data center EAPS domain. You need to do this configuration on the core and data center L2 switches.

   Core-Switch#1 control VLAN configuration:

   ```
   * CORE-SWITCH#1.1 # create vlan control-6
   * CORE-SWITCH#1.2 # configure vlan control-6 tag 4006
   * CORE-SWITCH#1.4 # configure vlan control-6 add port 4:1,4:2 tagged
   * CORE-SWITCH#1.5 # configure eaps e5-domain add control vlan control-6
   * CORE-SWITCH#1.6 # enable eaps e6-domain
   ```

   Data center-Switch#1 control VLAN configuration:

   ```
   * DC-SWITCH#1.1 # create vlan control-6
   * DC-SWITCH#1.2 # configure vlan control-6 tag 4006
   * DC-SWITCH#1.4 # configure vlan control-6 add port 49,50 tagged
   * DC-SWITCH#1.5 # configure eaps e6-domain add control vlan control-6
   * DC-SWITCH#1.6 # enable eaps
   * DC-SWITCH#1.7 # enable eaps e6-domain
   ```

   Dat acenter-Switch#2 control VLAN configuration:

   ```
   * DC-SWITCH#2.1 # create vlan control-6
   * DC-SWITCH#2.2 # configure vlan control-6 tag 4006
   * DC-SWITCH#2.4 # configure vlan control-6 add port 49,50 tagged
   * DC-SWITCH#2.5 # configure eaps e6-domain add control vlan control-6
   * DC-SWITCH#2.6 # enable eaps
   * DC-SWITCH#2.7 # enable eaps e6-domain
   ```

Create and Configure the Data Center EAPS Protected VLANs

1. Create the *EAPS* protected *VLAN* for the data center domain.
2. Configure the 802.1q tag and ports for the protected VLANs.

   Because each server is dual-homed to each data center switch, add a VLAN port on each switch for each server.
3. Configure the protected VLAN as part of the EAPS domain. Do this configuration on the core and data center switches.

   Core-Switch#1 protected VLAN configuration:

   ```
   * CORE-SWITCH#1.7 # create vlan srvfarm-1
   * CORE-SWITCH#1.8 # configure vlan srvfarm-1 tag 1000
   * CORE-SWITCH#1.9 # configure vlan srvfarm-1 add port 4:1,4:2 tagged
   * CORE-SWITCH#1.10 # configure eaps e6-domain add protect vlan srvfarm-1
   ```

Data center-Switch#1 protected VLAN configuration:

```
* DC-SWITCH#1.8 # create vlan srvfarm-1
* DC-SWITCH#1.9 # configure vlan srvfarm-1 tag 1000
* DC-SWITCH#1.10 # configure vlan srvfarm-1 add port 49,50 tagged
* DC-SWITCH#1.11 # configure vlan srvfarm-1 add port 1 untagged
* DC-SWITCH#1.12 # configure eaps e5-domain add protect vlan srvfarm-1
```

Data center-Switch#2 protected VLAN configuration:

```
* DC-SWITCH#2.8 # create vlan srvfarm-1
* DC-SWITCH#2.9 # configure vlan srvfarm-1 tag 1000
* DC-SWITCH#2.10 # configure vlan srvfarm-1 add port 49,50 tagged
* DC-SWITCH#2.11 # configure vlan srvfarm-1 add port 1 untagged
* DC-SWITCH#2.12 # configure eaps e5-domain add protect vlan srvfarm-1
```

Configuring an IP Address and OSPF on the Backbone VLAN

1. Configure an IP address and enable IP forwarding (routing) on the backbone protected *VLAN*.
2. *OSPF* is configured and because an OSPF broadcast network is used, configure the designated router and backup designated router for each VLAN.

   Since it is not performing L2 blocking, configure the *EAPS* transit core switch as the DR by using a higher OSPF priority value. The EAPS master core switch is configured as the BDR. The aggregation transit switches need not perform DR/BDR duties for the backbone VLAN, so their OSPF priority is configured at 0 to force ODR behavior.

   Core-Switch#1 OSPF configuration:

```
* CORE-SWITCH#1.12 # configure vlan backbone ipaddress 192.168.1.1/24
* CORE-SWITCH#1.13 # enable ipforwarding vlan backbone
* CORE-SWITCH#1.14 # configure ospf routerid 192.168.1.1
* CORE-SWITCH#1.15 # configure ospf add vlan backbone area 0.0.0.0
* CORE-SWITCH#1.16 # configure ospf vlan backbone priority 110
* CORE-SWITCH#1.17 # enable ospf
```

   Core-Switch#2 OSPF configuration:

```
* CORE-SWITCH#2.12 # configure vlan backbone ipaddress 192.168.1.2/24
* CORE-SWITCH#2.13 # enable ipforwarding vlan backbone
* CORE-SWITCH#2.14 # configure ospf routerid 192.168.1.2
* CORE-SWITCH#2.15 # configure ospf add vlan backbone area 0.0.0.0
* CORE-SWITCH#2.16 # configure ospf vlan backbone priority 100
* CORE-SWITCH#2.17 # enable ospf
```

   Agg-Switch#1 OSPF configuration:

```
* AGG-SWITCH#1.11 # configure vlan backbone ipaddress 192.168.1.3/24
* AGG-SWITCH#1.12 # enable ipforwarding vlan backbone
* AGG-SWITCH#1.13 # configure ospf add vlan backbone area 0.0.0.0
* AGG-SWITCH#1.14 # configure ospf vlan backbone priority 0
```

   Agg-Switch#2 OSPF configuration:

```
* AGG-SWITCH#2.11 # configure vlan backbone ipaddress 192.168.1.4/24
* AGG-SWITCH#2.12 # enable ipforwarding vlan backbone
* AGG-SWITCH#2.13 # configure ospf add vlan backbone area 0.0.0.0
* AGG-SWITCH#2.14 # configure ospf vlan backbone priority 0
```

Verifying EAPS and OSPF Configuration Status

1. Verify that the backbone *EAPS* domain and *OSPF* configuration is correct.
2. Confirm that the OSPF neighbor adjacencies and DR/BDR/ODR status are correct. Verify this status on both aggregate switches.

   Core-Switch#1 EAPS and OSPF status example:

```
* CORE-SWITCH#1.18 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 1
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e5-domain Links-Up T Y 2:1 2:4 control-5 (4005) 1
--------------------------------------------------------------------------------
* CORE-SWITCH#1.19 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
192.168.1.3 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.3 backbone
192.168.1.4 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.4 backbone
192.168.1.2 100 FULL /BDR 00:05:23:17/00:00:00:09 192.168.1.2 backbone
```

   Core-Switch#2 EAPS and OSPF status example:

```
* CORE-SWITCH#2.18 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 1
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e5-domain Complete T Y 2:1 2:4 control-5 (4005) 1
--------------------------------------------------------------------------------
* CORE-SWITCH#2.19 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
192.168.1.3 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.3 backbone
192.168.1.4 0 2WAY /DROTHER00:05:23:17/00:00:00:07 192.168.1.4 backbone
192.168.1.1 110 FULL /DR 00:05:23:17/00:00:00:09 192.168.1.1 backbone
```

   Agg-Switch#1 EAPS and OSPF status example:

```
* AGG-SWITCH#1.15 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 5
# EAPS domain configuration :
```

```
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e1-domain Links-Up T Y 1:1 2:1 control-1 (4001) 2
e2-domain Links-Up T Y 1:4 2:1 control-2 (4002) 2
e3-domain Links-Up T Y 3:1 2:1 control-3 (4003) 2
e4-domain Links-Up T Y 3:2 2:1 control-4 (4004) 2
e5-domain Links-Up T Y 2:4 2:6 control-5 (4005) 1
--------------------------------------------------------------------------------
* AGG-SWITCH#1.16 # show ospf neighbor
Neighbor ID Pri State Up/Dead Time Address Interface
192.168.1.1 110 FULL /DR 00:00:28:51/00:00:00:01 192.168.1.1 backbone
192.168.1.2 100 FULL /BDR 00:00:28:51/00:00:00:01 192.168.1.2 backbone
192.168.1.4 0 2WAY /DROTHER00:05:45:40/00:00:00:03 192.168.1.4 backbone
172.16.1.2 100 FULL /BDR 00:18:01:08/00:00:00:03 172.16.3.2 orange-1
172.16.1.2 100 FULL /BDR 00:18:01:08/00:00:00:03 172.16.4.2 red-1
172.16.1.2 100 FULL /BDR 00:17:54:17/00:00:00:03 172.16.1.2 green-1
172.16.1.2 100 FULL /BDR 00:17:54:07/00:00:00:03 172.16.2.2 purple-1
```

Agg-Switch#2 EAPS and OSPF status example:

```
* AGG-SWITCH#2.15 # show eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: On
EAPS Display Config Warnings: On
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 5
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain State Mo En Pri Sec Control-Vlan VID Count
--------------------------------------------------------------------------------
e1-domain Complete M Y 2:1 1:1 control-1 (4001) 2
e2-domain Complete M Y 2:1 1:4 control-2 (4002) 2
e3-domain Complete M Y 2:1 3:1 control-3 (4003) 2
e4-domain Complete M Y 2:1 3:2 control-4 (4004) 2
e5-domain Links-Up T Y 2:4 2:6 control-5 (4005) 1
--------------------------------------------------------------------------------
* AGG-SWITCH#2.16 # show ospf neighbor
Interface
192.168.1.1 110 FULL /DR 00:00:28:51/00:00:00:01 192.168.1.1 backbone
192.168.1.2 100 FULL /BDR 00:00:28:51/00:00:00:01 192.168.1.2 backbone
192.168.1.3 0 2WAY /DROTHER00:05:45:40/00:00:00:03 192.168.1.3 backbone
172.16.1.1 110 FULL /DR 00:18:01:08/00:00:00:03 172.16.3.1 orange-1
172.16.1.1 110 FULL /DR 00:18:01:08/00:00:00:03 172.16.4.1 red-1
172.16.1.1 110 FULL /DR 00:17:54:17/00:00:00:03 172.16.1.1 green-1
172.16.1.1 110 FULL /DR 00:17:54:07/00:00:00:03 172.16.2.1 purple-1
```

CFM Support in EAPS

ExtremeXOS provides Connectivity Fault Management (CFM) support within EAPS protocol.

CFM reports fault connectivity failures to *EAPS*, and EAPS communicates with the CFM process to set up point-to-point DOWN MEPs (Management Endpoints) to monitor link connectivity. The CFM module notifies EAPS of any link-connectivity issues, and triggers EAPS to take necessary action.

802.1ag CFM supports link monitoring. It does this by sending out PDUs at designated transmit intervals. If the CFM fails to receive PDUs, it assumes the link is out of service, and notifies its clients. In this instance, EAPS acts as a CFM client.

First, you will create a down MEP within the CFM CLI. Configure the CLI to create a MEP group that associates this down MEP with a remote MEP (RMEP). There is a 1:1 relationship between a port and the down MEP, and as such, each MEP group is tied to a single port. Using the EAPS CLI, you can add the MEP groups you wish to monitor. For each MEP group added to EAPS, EAPS will receive UP/DOWN notifications from CFM when CFM detects a MEP state change for that group. Each MEP group corresponds to an EAPS ring port. Notifications from those MEP groups that are inadvertently added—that do not correspond to an EAPS ring port—are ignored in EAPS.

The CFM configuration is independent of EAPS, and MEPs and MEP groups may use different VLANs other than the EAPS control *VLAN* to monitor links.

When EAPS receives a CFM notification that the link failed, EAPS blocks that port on all of the EAPS control VLANs. This prevents EAPS control PDUs from being hardware forwarded on the link, in case the link is still up. Any EAPS PDUs that are received on a CFM failed port are dropped in EAPS. It is generally recommended to have the CCM interval as short as possible to detect link failures faster, which improves convergence time.

Configuring EAPS for CFM Support

- Use the following command to configure *EAPS* for CFM support:

For additional configuration details for CFM support, refer to Configuring CFM on page 1290.

Binding to a MEP Group

- To bind to a MEP Group, use the following command:

  ```
  configure eaps cfm [add | delete] group group_name
  ```

  This command notifies CFM that EAPs is interested in notifications for this MEP and RMEP pair. This MEP should already be bound to a physical port, so when notification is received, *EAPS* associates that notification with a ring-port failure.

Create MPs and the CCM Transmission Interval

Within an MA, you configure the following MPs:

- Maintenance end points (MEPs), which are one of the following types:
- ◦ UP MEPs—transmit CCMs and maintain CCM database
  ◦ DOWN MEPs—transmit CCMs and maintain CCM database
- Maintenance intermediate points (MIPs)—pass CCMs through

Each MEP must have an ID that is unique for that MEP throughout the MA.

- To configure UP and DOWN MEPs and its unique MEP ID, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* [**`ports`** *`port_list`* **`add`** [[**`end-point`** [**`up`**|**`down`**] *`mepid`* {**`group`** *`group_name`*}] | [**`intermediate-point`**]]

- To change the MEP ID on an existing MEP, use the following command:

  `configure cfm domain` *`domain-name`* **`association`** *`association_name`* **`ports`** *`port_list`* **`end-point`** [**`up`** | **`down`**] **`mepid`** *`mepid`*

- To delete UP and DOWN MEPs, use the following command:

  `configure cfm domain` *`domain-name`* **`association`** *`association_name`* **`ports`** *`port_list`* `delete` **`end-point`** [**`up`** | **`down`**] **`intermediate-point`**

- To configure a MIP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* [**`ports`** *`port_list`* **`add`** [[**`end-point`** [**`up`**|**`down`**] *`mepid`* {**`group`** *`group_name`*}] | [**`intermediate-point`**]]

- To delete a MIP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* [**`ports`** *`port_list`* **`delete`** [[**`end-point`** [**`up`**|**`down`**] *`mepid`* {**`group`** *`group_name`*}] | [**`intermediate-point`**]]

- To configure the transmission interval for the MEP to send CCMs, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* {**`ports`** *`port_list`* **`end-point`** [**`up`** | **`down`**]} **`transmit-interval`** [**`3`**|**`10`**|**`100`**|**`1000`**|**`10000`**|**`60000`**|**`600000`**]

- To unconfigure the transmission interval for the MEP to send CCMs and return it to the default, use the following command:

  `unconfigure cfm domain` *`domain_name`* **`association`** *`association_name`* {**`ports`** *`port_list`* **`end-point`** [**`up`** | **`down`**]} **`transmit-interval`**

- To enable of disable a MEP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* **`end-point`** [**`up`** | **`down`**] [**`enable`** | **`disable`**]

Displaying EAPS MEP Group Bindings

- Display *EAPS* MEP group bindings with the command: `show eaps cfm groups`

```
# sh eaps cfm groups
-------------------------------------------------------------------------
MEP Group Name                  Status Port   MEP ID
-------------------------------------------------------------------------
eapsCfmGrp1                     Up     41     11
eapsCfmGrp2                     Up     31     12
```

Displaying EAPS Output Change

- Display *EAPS* output changes using the command `show eaps`

  The existing output places a ! next to a CFM monitored ring port if the CFM indicates the MEP group for that port is down.

```
# sh eaps
EAPS Enabled: Yes
EAPS Fast-Convergence: Off
EAPS Display Config Warnings: Off
EAPS Multicast Add Ring Ports: Off
EAPS Multicast Send IGMP Query: On
EAPS Multicast Temporary Flooding: Off
EAPS Multicast Temporary Flooding Duration: 15 sec
Number of EAPS instances: 1
# EAPS domain configuration :
--------------------------------------------------------------------------------
Domain      State        Mo  En  Pri    Sec    Control-Vlan VID   Count Prio
--------------------------------------------------------------------------------
d2          Failed       M   Y   !41     31    v2           (101 ) 1     N
--------------------------------------------------------------------------------
Flags : (!) CFM Down
```

Configuration Example

Below is a sample configuration of CFM support in *EAPS*:

```
switch 1 # sh configuration cfm
#
# Module dot1ag configuration.
#
create cfm domain string "MD1" md-level 6
configure cfm domain "MD1" add association string "MD1v1" vlan "v1"
configure cfm domain "MD1" add association string "MD1v2" vlan "v2"
configure cfm domain "MD1" association "MD1v1" ports 17 add end-point down 6
configure cfm domain "MD1" association "MD1v1" ports 23 add end-point down 5
configure cfm domain "MD1" association "MD1v2" ports 31 add end-point down 13
configure cfm domain "MD1" association "MD1v1" ports 17 end-point down add group
"eapsCfmGrp1"
configure cfm domain "MD1" association "MD1v1" ports 23 end-point down add group
"eapsCfmGrp2"
configure cfm domain "MD1" association "MD1v2" ports 31 end-point down add group
"eapsCfmGrp3"
configure cfm group "eapsCfmGrp1" add rmep 2
configure cfm group "eapsCfmGrp2" add rmep 4
configure cfm group "eapsCfmGrp3" add rmep 12
switch 2 # sh configuration "eaps"s
#
# Module eaps configuration.
#
enable eaps
create eaps d1
configure eaps d1 mode transit
configure eaps d1 primary port 17
configure eaps d1 secondary port 23
enable eaps d1
create eaps d2
configure eaps d2 mode transit
configure eaps d2 primary port 31
configure eaps d2 secondary port 23
enable eaps d2
configure eaps d1 add control vlan v1
configure eaps d1 add protected vlan pv1
configure eaps d2 add control vlan v2
```

```
configure eaps d2 add protected vlan pv2
create eaps shared-port 23
configure eaps shared-port 23 mode partner
configure eaps shared-port 23 link-id 100
configure eaps cfm add group eapsCfmGrp1
configure eaps cfm add group eapsCfmGrp2
configure eaps cfm add group eapsCfmGrp3
```

Limitations

The following features of Change-of-Authorization (RFC5176) are not implemented in ExtremeXOS:

- Reverse Path Forwarding Check—Typically this is used in a proxy scenario. This check is used to determine if the IP address indicated by the *RADIUS (Remote Authentication Dial In User Service)* attributes is a routable destination address for a request sent by the switch software.
- IPSEC encryption—End-to-end encryption of both the RADIUS requests and responses.
- Disconnect-Request and Change-of-Authorization packets identifying sessions with anything other than the Calling-Station-Id attribute containing a properly formatted MAC address. In addition to the Calling-Station-ID attribute, you can also use a NAS-Port attribute, which indicates the index of the specific port the session is connected to.
- Acct-Session-Id attribute—This is an alternate means of session identification. Sessions are currently uniquely identified by port and MAC address pair.
- Retransmissions of Disconnect-Request or Change-of-Authorization ACK and NAK packets—Retransmissions of packets is the responsibility of the device initiating the dynamic authorization transactions.

Platforms Supported

All ExtremeXOS platforms support this feature; however, not all platforms support hardware-based CFM.

Platforms with no hardware-based CFM support are limited to software-based CFM transmit intervals of 100 ms or higher. Hardware-based intervals can go as low as 3.3 ms.

ERPS

ERPS Overview on page 604
Supported ERPS Features on page 606
G.8032 Version 2 on page 606
Configuring ERPS on page 613
Sample Configuration on page 616
Debugging ERPS on page 617
ERPS Feature Limitations on page 617

This chapter provides an overview to ERPS, and discusses various ERPS features. The chapter also offers configuration details, provides configuration examples, and shows you how to debug ERPS.

ERPS Overview

The basic concept of G.8032/*ERPS (Ethernet Ring Protection Switching)* is that traffic may flow on all links of a ring network except on one link called the Ring Protection Link (RPL).

The RPL owner is the node that blocks the RPL, and the other node of the RPL is called the *RPL neighbor node*. All other nodes are called *non-RPL nodes*. When a link fails, the RPL owner unblocks the RPL to allow connectivity to the nodes in the ring. The G.8032/ERPS rings utilize a channel (dedicated path) for carrying their control traffic which is the R-APS messages (Ring Automatic Protection Switching).

The ring protection architecture relies on the existence of an APS protocol to coordinate ring protection actions around an Ethernet ring, as shown in Figure 61.

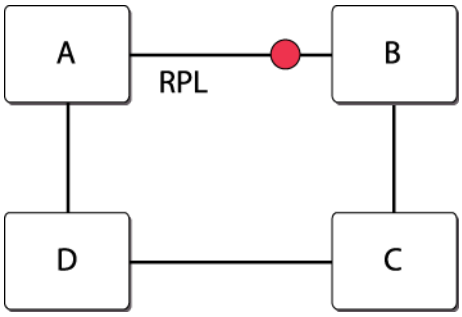


Figure 61: Simple Ring with RPL, RPL Owner, RPL Neighbor, and Non-RPL Nodes

More complex topologies include ladder ring networks which are called *sub-rings* in G.8032 terminology. In these networks, there could exist one or more rings and

sub-rings which complete their connectivity through the interconnected nodes of the ring(s). Multiple ladder networks are supported only if the following conditions are met:

- R-APS channels are not shared across Ethernet ring interconnections.
- On each ring port, each traffic channel and each R-APS channel are controlled by the Ethernet Ring Protection (ERP) Control process of only one Ethernet ring.
- Each major ring or sub-ring must have its own RPL.

Note
One important aspect of sub-rings is that they complete their channel through the virtual channel (when using the virtual channel mode), which can span the network and cross the sub-ring boundaries. This entails that the virtual channel is provisioned on all the nodes it spans across.

In Figure 62, the ring comprises nodes A, B, C, and D with links A–B, B–C, C–D, and D–A while the control channel for this ring has its own dedicated *VLAN (Virtual LAN)*. The sub-ring consists of nodes D, F, E, and C with links D–F, F–E, and E–C. D and C are interconnected nodes. The channel for the sub-ring spans the links C–E, E–F, and F–D and their nodes while the virtual channel comprises the links D-A, A-B, B-C and D–C and their nodes. This means that the virtual channel for the sub-ring needs to not only exist on the interconnected nodes, but also on the nodes A and B.

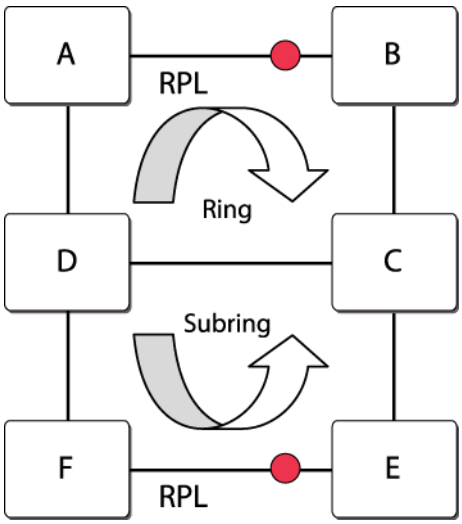


Figure 62: Ring and Sub-ring Network

When using G.8032 in networks, take care to design the virtual channel paths, since the VLAN provisioning has to exist on all the nodes through which the virtual channel can pass and which is solely dedicated to the sub-ring in question.

Sub-ring topology changes may impact flow forwarding over the domain of the other (interconnected) network, as such topology change events are signaled to the domain of the other network using the Topology Change signal.

Supported ERPS Features

The following are the *ERPS* features supported in the current release:

- G.8032 version 1 support.
- G.8032 version 2 support with a restricted VC option.
- Revertive mode support for version 1 and 2.
- Basic interoperability with *EAPS (Extreme Automatic Protection Switching)* with G.8032 acting as an access ring. Flush notifications will be sent Link monitoring using CFM or native local link monitoring methods.
- Support for hardware accelerated CFM in specific platforms that have this capability.
- G.8032 version 2 with no Virtual Channel support.
- Support for attaching to a CFM DOWN-MEP configured external to ERPS.

G.8032 Version 2

The concept of sub-rings is introduced to add multiple rings to the main ring. A sub-ring is an incomplete ring that completes its path through the main ring or other sub-rings. The control path for the sub-ring completes either through the implementation of a virtual channel, or by changing the flow of control packets in the sub-rings. Virtual channels are supported through the use of the sub-rings control channel being configured as a data *VLAN* in the main ring.

You can configure the sub-ring in “no virtual channel” mode, where the control path for the sub-ring is through all the nodes of the sub-ring (including the RPL owner and neighbor). You must be careful, however, to avoid using the sub-ring’s control channel across the main ring because that will cause a loop. ExtremeXOS supports the use of CFM, in conjunction with Manual Switch (MS), to protect the sub-rings against multiple failures in the main ring.

CFM Link Monitoring

To enable CFM to report link events, the link must first be registered with CFM. *ERPS* acts as a client of CFM and creates the required Management Entity Points (MEPs). For G.8032 v1/v2 implementation:

It is generally recommended to have the CCM interval as short as possible to detect link failures faster, which improves convergence time.

- Creating a DOWN-MEP is by creating the DOWN-MEP with the CFM commands, and then assigning a group name to it. This group can then be associated to the ERPS ring.

Here is an example:

```
switch # sh  cfm
Domain: "erps_6",  MD Level: 6
   Association: "erps_MA_100", Destination MAC Type:  Multicast, VLAN "v2" with 2 cfm
ports
   Transmit Interval: 1000 ms
     port 27; Down End Point, mepid: 11, transmit-interval: 10000 ms (configured),
```

```
        MEP State: Enabled, CCM Message: Enabled,  Send SenderId TLV: Disabled
     port 37; Down End Point, mepid: 21,
        transmit-interval: 10000 ms (configured),
        MEP State: Enabled, CCM Message: Enabled, Send SenderId TLV: Disabled
   Association: "erps_MA_100", Destination MAC Type: Multicast, VLAN "v2" with 2 cfm
ports
   Transmit Interval: 1000 ms
Total Number of Domain             : 1
Total Number of Association        : 2
Total Number of Up MEP             : 0
Total Number of Down MEP           : 2
Total Number of MIP                : 0
Total Number of Number of CFM port : 4
Total Number of VPLS MIP(Static/Up): 0 / 0

switch # show cfm detail
Domain/       Port     MP  Remote End-Point  Remote End-Point   MEP   Life       Flags
Association                  MAC Address       IP Address       ID    time   Age
================================================================================
erps_6
  erps_MA_100  27      DE     00:04:96:34:e3:43  0.0.0.0        10    35000  4430   DM

               37      DE     00:04:96:27:fb:7b 0.0.0.0         20    35000  2790   DM
================================================================================
Maintenance Point: (UE) Up End-Point, (DE) Down End-Point
Flags: S - Static  Entry D - Dynamic Entry
          CCM Destination MAC: (U) Unicast (M)  Multicast
NOTE: The Domain and Association names are truncated to 13 characters, Lifetime and Age
are in milliseconds.
================================================================================

Total Number of Dynamic Up RMEP      : 0
Total Number of Dynamic Down RMEP    : 2
Total Number of Active Static RMEP   : 0
Total Number of Inactive Static RMEP : 0
```

Note
You must configure a remote MEP-ID for the local MEPs so that a specific association can be maintained between the two ends.

Limitations

CFM PDU transmit intervals are limited by the supported limits of CFM module. Platforms that do not support CFM in hardware are limited to a minimum interval of 100 ms. The maximum number of down MEPs is limited by the CFM module. This is as low as 32 MEPs in some platforms.

Revertive and Non-revertive Mode

In the revertive mode, you can revert back to the RPL being blocked once the Signal Fault has cleared. In non-revertive mode, the SF remains blocked even after the fault clears. Reversion is handled in the following way:

- The reception of an R-APS No Request (NR) message causes the RPL owner node to start the wait-to-restore (WTR) timer.
- The WTR timer is cancelled if, during the WTR period, a request with a higher priority than NR is accepted by the RPL owner node, or is declared locally at the RPL owner node.

- When the WTR timer expires, without the presence of any other higher priority request, the RPL owner node initiates reversion by blocking its traffic channel over the RPL, transmitting an R-APS (NR, RB) message over both ring ports, informing the Ethernet ring that the RPL is blocked, and performing a flush *FDB (forwarding database)* action. The *ERPS* Ring will be in the idle state.
- The acceptance of the R-APS (NR, RB) message causes all Ethernet ring nodes to unblock any blocked non-RPL link that does not have an SF condition. If it is an R-APS (NR, RB) message without a DNF indication, all Ethernet ring nodes perform a necessary flush FDB action.

In non-revertive operation, the Ethernet ring does not automatically revert when all ring links and Ethernet ring nodes have recovered and no external requests are active. Non-revertive operation is handled in the following way:

- The RPL owner node does not generate a response on reception of an R-APS (NR) messages.
- When other healthy Ethernet ring nodes receive the NR (node ID) message, no action is taken in response to the message.

Force Switch

In the absence of any failure in the ring network, an operator-initiated Force Switch (FS) results in the RPL getting unblocked, and the node on which the FS has been issued is blocked. This condition is indicated by the transmission of R-APS FS messages, which are continuous until this condition is unconfigured. Two or more Forced Switches are allowed in the Ethernet ring, but this may cause the segmentation of an Ethernet ring. It is the responsibility of the operator to prevent this effect if it is undesirable.

You can remove a Forced Switch condition by issuing a clear command to the same Ethernet ring node where the Forced Switch is presented (`configure erps` *`ring-name`* **`dynamic-state`** [**`force-switch`** | **`manual-switch`** | **`clear`**] **`port`** *`slot:port`*). The clear command removes existing local operator commands and triggers reversion in case the Ethernet ring is in revertive behavior. The Ethernet ring node where the Forced Switch was cleared continuously transmits the R-APS (NR) message on both ring ports, informing that no request is present at the Ethernet ring node.

Clear Command

The clear command is described in the *ITU-T G.8032* (*Standard for ERPS*). This command allows the ERPS ring to administratively return (revert) to idle state and trigger (on the owner) the transmission of NR,RB messages that return all other nodes back to the idle state from the pending state. If it is an R-APS (NR, RB) message without a DNF indication, all Ethernet ring nodes perform a necessary flush FDB. In the revertive-mode of operation, after the link failure is cleared in the ring, the traffic channel reverts (the RPL owner node blocks its RPL port and transmits an R-APS (NR, RB) message in both directions, repeatedly) after the expiration of a WTR timer. With the clear command, the traffic channel can be made to revert without waiting for the WTR to expire. When you issue a clear command for non-revertive mode at the RPL owner node, the non-revertive operation is cleared and the traffic channel is reverted.

Upon receiving an R-APS (NR, RB) message, any blocking Ethernet ring node should unblock its non-failed ring port.

To issue the clear command, use the command:

```
configure erps ring-name dynamic-state [force-switch | manual-switch |
clear] port slot:port
```

Manual Switch

Manual Switch is similar to the Force Switch except that only one Manual Switch is allowed for an Ethernet ring. The processing of which node retains the Manual Switch is based on the priority table and the node state. However only one Manual Switch is retained at the end for the ring.

Clearing the Manual Switch is done similar to the Force Switch.

Virtual Channel for Sub-rings

While the standard describes how the sub-rings can function with a virtual channel, in this implementation sub-rings will function only with the presence of virtual channels.

Channel Blocking

The R-APS control channel is blocked, as is traffic on the blocked ports for the control traffic entering on one ring port and getting forwarded to the other ring port. However, locally generated or delivered control traffic on the blocked port is supported.

Traffic Blocking

Traffic is always blocked for the protected VLANs on the blocked ports of the ring/sub-ring in a G.8032 network.

Signal Failure and Recovery

In the absence of a higher priority request in the node, the following Signal Failure (SF) actions are taken.

- An Ethernet ring node detecting an SF condition on one of its ring ports blocks the traffic channel and R-APS channel on the failed ring port.
- The Ethernet ring node detecting an SF condition transmits an R-APS message indicating SF on both ring ports. The R-APS (SF) message informs other Ethernet ring nodes of the SF condition. R-APS (SF) messages are continuously transmitted by the Ethernet ring node detecting the SF condition while this condition persists. (The Periodic timer determines the interval of sending the SF after the first three.) For sub-ring interconnection nodes, the R-APS (SF) message is transmitted on the R-APS channel of the Sub-Ring port.
- Assuming the Ethernet ring node was in an idle state before the SF condition occurred, upon detection of this SF condition the Ethernet ring node triggers a local *FDB* flush.
- An Ethernet ring node accepting an R-APS (SF) message unblocks any blocked ring port that does not have an SF condition. This action unblocks the traffic channel on the RPL.

- An Ethernet ring node accepting an R-APS (SF) message stops transmission of other R-APS messages.
- An Ethernet ring node accepting an R-APS (SF) message without a DNF indication performs a flush FDB.

An Ethernet ring node that has one or more ring ports in an SF condition (upon detection of clearance of the SF condition) keeps at least one of these ring ports blocked for the traffic channel and for the R-APS channel, until the RPL is blocked as a result of Ethernet ring protection reversion, or until there is another higher priority request (for example, an SF condition) in the Ethernet ring. An Ethernet ring node that has one ring port in an SF condition, and detects clearing of this SF condition, continuously transmits the R-APS (NR) message with its own Node ID as the priority information over both ring ports, informing that no request is present at the Ethernet ring node and initiates a guard timer as described in sub-clause 10.1.5. Another recovered Ethernet ring node (or Nodes) holding the link block receives the message and compares the Node ID information with its own Node ID. If the received R-APS (NR) message has the higher priority, the Ethernet ring node unblocks its ring ports. Otherwise, the block remains unchanged. There is only one link with one-end block. The Ethernet ring nodes stop transmitting R-APS (NR) messages when they accept an R-APS (NR, RB), or when another higher priority request is received

R-APS Control MAC

The R-APS messages use the MAC address range allocated within ITU OUI for G.8032 R-APS communication, which is 01:19:A7:00:00:ring-id or 01:19:A7:00:00:01; whereas ring-id ranges from 1 to 239. ExtremeXOS supports both type of R-APS control MACs. You can use either “01:19:A7:00:00:01 (auto)” or “01:19:A7:00:00:ring-id (default)” as a R-APS control MAC using the following command: `configure` **`erps`** *`ring-name`* **`control-mac`** [**`auto`** | **`default`**].

Note that this command is only applicable on ring instances created with user-defined ring-id.

Control MAC by Default

- If the ERPS ring is created with a user-defined ring-id, then the control-mac used by the ring will be “01:19:A7:00:00:ring-id (control-mac type auto)” by default.
- If the ERPS ring is created without a user-defined ring-id, then the control-mac used by ring will be “01:19:A7:00:00:01 (control-mac type default)” by default.
- If you are upgrading ExtremeXOS from an older version (without ring-id–based control MAC support) to newer version (with ring-id–based control MAC support), all configurations are retained and unique ring IDs are assigned to those ERPS ring instances by default. R-APS control MAC used by these rings IS “01:19:A7:00:00:01 (default)” by default.

Note that the control MAC configuration as auto is not applicable on such ring instances.

Note
Be advised that the ITU_T standard does not have the mechanism to recognize configuration mismatches across the nodes, especially with the ring-ID/control MAC configuration. Therefore, you must make sure that the all of the ring instances on all nodes within the same physical ring have the same ring ID and control MAC.

Timers

This section discusses the various timers associated with *ERPS*.

Guard Timer

The guard timer is used to prevent Ethernet ring nodes from acting upon outdated R-APS messages, and to prevent the possibility of forming a closed loop. The guard timer is activated whenever an Ethernet ring node receives an indication that a local switching request has cleared (i.e., local clear SF, clear). The guard timer can be configured in 10 ms steps, between 10 ms and two seconds, with a default value of 500 ms. This timer period should be greater than the maximum expected forwarding delay in which an R-APS message traverses the entire ring. The longer the period on the guard timer, the longer an Ethernet ring node is unaware of new or existing relevant requests transmitted from other Ethernet ring nodes, and is unable to react to them.

A guard timer is used in every Ethernet ring node. Once a guard timer is started, it expires by itself. While the guard timer is running, any received R-APS Request/State and Status information is blocked and not forwarded to the Priority Logic. When the guard timer is not running, the R-APS Request/State and Status information is forwarded unchanged.

Hold-off Timer

W hen a new defect, or more severe defect occurs (new SF), this event is not be reported immediately to protection switching if the provisioned hold-off timer is a non-zero value. Instead, the hold-off timer is started. When the hold-off timer expires, the trail that started the timer is checked to see if a defect still exists. If one does exist, that defect is reported to protection switching. The suggested range of the hold-off timer is 0 to 10 seconds in steps of 100 ms with an accuracy of ±5 ms. The default value for a hold-off timer is 0 seconds.

Delay Timers

In revertive mode, the wait-to-restore (WTR) timer is used to prevent frequent operation of the protection switching caused by intermittent signal failure defects. The wait-to-block (WTB) timer is used when clearing Forced Switch and Manual Switch commands. As multiple Forced Switch commands are allowed to coexist in an Ethernet ring, the WTB timer ensures that clearing of a single Forced Switch command does not trigger the re-blocking of the RPL. When clearing a Manual Switch command, the WTB timer

prevents the formation of a closed loop due to a possible timing anomaly where the RPL owner node receives an outdated remote MS request during the recovery process.

Interoperability with EAPS

A basic G.8032 interoperability with EAPS is supported wherein the G.8032 ring is configured as sub-ring ring interacting with the backbone EAPS ring.

When topology changes occur in the G.8032 ring, this initiates a ring flush in the EAPS ring. This uses the same methodology that EAPS uses today when STP requires a flush. EAPS then flushes all the VLANs in its domains.

When a failure occur in the G.8032 ring, the protocol handles it and open up the RPL link. A flush message is propagated to the EAPS ring if configured to do so using the `configure erps` *`ring-name`* [**`add`** |**`delete`**] **`topology-change`***`ring-list`* command.

Sample Configuration

Here is a sample configuration of the *ERPS* feature:

```
create vlan cv1
config vlan cv1 tag 10
config vlan cv1 add port 5,6 tagged

create vlan pv1
config vlan pv1 tag 1000
config vlan pv1 add port 1
config vlan pv1 add port 5,6 tagged

create erps ring1
configure erps ring1 add ring-ports east 5
configure erps ring1 add ring-ports west 6
configure erps ring1 add control “cv1”
configure erps ring1 add protected vlan “pv1”
configure erps ring1 add protection-port 5
configure erps ring1 revert enabled wait-to-restore 500
enable erps r1
enable erps
```

CFM DOWN-MEP Configuration to Provide Link Monitoring/Notifications

```
create cfm domain string "MD3" md-level 3
configure cfm domain "MD3" add association string "MD3vsub1" vlan "vsub1"
configure cfm domain "MD3" association "MD3vsub1" ports 20 add end-point down 14
configure cfm domain "MD3" association "MD3vsub1" ports 24 add end-point down 13
configure cfm domain "MD3" association "MD3vsub1" ports 20 end-point down add group
"erpsDn1"
configure cfm domain "MD3" association "MD3vsub1" ports 24 end-point down add group
"erpsDn2"
configure cfm group "erpsDn1" add rmep 15
configure cfm group "erpsDn2" add rmep 12
configure erps subring1 cfm port east add group erpsDn2
configure erps subring1 cfm port west add group erpsDn1
```

Sub-ring Configuration

First, configure a main ring on the Interconnected node:

```
create erps main-ring1
configure erps main-ring1 add ring-ports east 5
configure erps main-ring1 add ring-ports west 6
configure erps ring1 add control "cv1"
```

Next, configure a sub-ring on the interconnected node:

```
create erps sub-ring1
configure erps sub-ring1 add ring-ports east 10
configure erps sub-ring1 add control "subv1"
configure erps main-ring1 add sub-ring sub-ring1
enable erps main-ring1
enable erps sub-ring1
```

Virtual Channel for Sub-ring

```
configure vlan subv1 add port 5 6 tagged
configure main-ring1 add protected vlan subv1
```

No Virtual Channel for Sub-ring

```
configure erps subring1 subring-mode no-virtualChannel
```

Configuring ERPS

ERPS Version 1 Commands

- To create or delete an *ERPS* ring, use the following commands:

  `create erps` *ring-name* {**ring-id** *ring_id*}

  `delete erps` *ring-name*

- To add or delete a control *VLAN* on the ERPS ring, use the following commands:

  `configure erps` *ring-name* **add control** {**vlan**} *vlan_name*

  `configure erps` *ring-name* **delete control** {**vlan**} *vlan_name*

- To add or delete a protected VLAN on the ERPS ring, use the following commands:

  `configure erps` *ring-name* **add protected** {**vlan**} *vlan_name*

  `configure erps` *ring-name* **delete protected** {**vlan**} *vlan_name*

- To add ring ports on the ERPS ring (this ring's ports connect the switch to the ERPS ring), use the following command:

  `configure erps` *ring-name* **ring-ports** [**east** | **west**] *port*

- To add ring ports on the ERPS ring, use the following command:

  `configure erps` *ring-name* **ring-ports** [**east** | **west**] *port*

- To configure CFM for ERPS ring ports, use the following command:

  `configure erps` *ring_name* **cfm port** [**east** | **west**] [**add** | **delete**] **group** *group_name*

- To delete ring ports on the ERPS ring, use the following command:

  ```
  unconfigure erps ring-name ring-ports west
  ```

- To add or delete RPL (ring protection link) owner configuration for the ERPS ring, use the following commands:

  ```
  configure erps ring-name protection-port port

  unconfigure erps ring-name protection-port
  ```

- To add or delete RPL (ring protection link) neighbor configuration for the ERPS ring, use the following commands:

  ```
  configure erps ring-name neighbor-port port

  unconfigure erps ring-name neighbor-port
  ```

- To add or delete ERPS revert operation along with the wait-to-restore time interval, use the following commands:

  ```
  configure {erps} ring-name revert [ enable | disable ]
  ```

- To configure the periodic timer, use the following command:

  ```
  configure {erps} ring-name timer periodic [ default | milliseconds ]
  ```

- To configure the guard timer, use the following command:

  ```
  configure {erps} ring-name timer guard [ default | milliseconds ]
  ```

- To configure the hold-off timer, use the following command:

  ```
  configure {erps} ring-name timer hold-off [ default | milliseconds ]
  ```

- To configure the wait-to-restore timer, use the following command:

  ```
  configure {erps} ring-name timer wait-to-restore [ default |
  milliseconds ]
  ```

- To rename the ERPS ring/sub-ring, use the following command:

  ```
  configure erps old-ring-name name new-ring-name
  ```

- To enable or disable ERPS, use the following commands:

  ```
  enable erps

  disable erps
  ```

- To enable or disable an existing ERPS ring/sub-ring, , use the following command:

  ```
  enable erps ring-name

  disable erps ring-name
  ```

- Run or clear force and manual switch triggers, or clear command to the ERPS ring/sub-ring.

  ```
  configure erps ring-name dynamic-state [force-switch | manual-switch |
  clear] port slot:port
  ```

- To display global information for ERPS, use the following command:

  ```
  show erps
  ```

- To display specific details about an ERPS ring, use the following command:

  ```
  show erps ring-name
  ```

- To display ERPS statistics, use the following command:

  ```
  show erps ring-name statistics
  ```

- To clear statistics on an ERPS ring, use the following command:

  ```
  clear counters erps ring-name
  ```

- To debug ERPS, use the following commands:

  ```
  debug erps [options]

  debug erps show ring-name
  ```

- To configure control MAC (either default or auto) on a particular ERPS ring instance, use the following command:

  ```
  configure erps ring-name control-mac [auto | default]
  ```

ERPS Version 2 Commands

- To set the rings to which to propagate topology change events, use the following command:

  ```
  configure erps ring-name [add | delete] topology-change ring-list
  ```

- To enable or disable the ability of *ERPS* to allow the topology-change bit to be set (to send out Flush events), , use the following commands:

  ```
  enable erps ring-name topology-change

  disable erps ring-name topology-change
  ```

- To add or delete a sub-ring to the main ring, use the following command:

  ```
  configure {erps} ring-name [add | delete] sub-ring-name sub_ring
  ```

- To configure the wait-to-block timer, use the following command:

  ```
  configure {erps} ring-name timer wait-to-block [ default |
  milliseconds]
  ```

- To add or delete an ERPS sub-ring to the *EAPS* domain, use the following commands:

  ```
  configure {erps} ring-name notify-topology-change {eaps} domain_name

  unconfigure {erps} ring-name notify-topology-change {eaps} domain_name
  ```

- To configure a wait-to-block timer, use the following command:

  ```
  configure {erps} ring-name timer wait-to-block [ default |
  milliseconds]
  ```

- To configure sub-ring mode, use the following command:

  ```
  configure erps ring_name subring-mode [no-virtualChannel |
  virtualChannel]
  ```

Note
If the ERPS configuration of RPL owner node is modified, you need to disable, and then enable, corresponding ERPS instance on all non-RPL nodes.

- To configure R-APS control MAC as either auto or default, use the following command:

  `configure` **`erps`** *`ring-name`* **`control-mac`** [**`auto`** | **`default`**]

Sample Configuration

Here is a sample configuration of the *ERPS* feature:

```
create vlan cv1
config vlan cv1 tag 10
config vlan cv1 add port 5,6 tagged

create vlan pv1
config vlan pv1 tag 1000
config vlan pv1 add port 1
config vlan pv1 add port 5,6 tagged

create erps ring1
configure erps ring1 add ring-ports east 5
configure erps ring1 add ring-ports west 6
configure erps ring1 add control “cv1”
configure erps ring1 add protected vlan “pv1”
configure erps ring1 add protection-port 5
configure erps ring1 revert enabled wait-to-restore 500
enable erps r1
enable erps
```

CFM DOWN-MEP Configuration to Provide Link Monitoring/Notifications

```
create cfm domain string "MD3" md-level 3
configure cfm domain "MD3" add association string "MD3vsub1" vlan "vsub1"
configure cfm domain "MD3" association "MD3vsub1" ports 20 add end-point down 14
configure cfm domain "MD3" association "MD3vsub1" ports 24 add end-point down 13
configure cfm domain "MD3" association "MD3vsub1" ports 20 end-point down add group
"erpsDn1"
configure cfm domain "MD3" association "MD3vsub1" ports 24 end-point down add group
"erpsDn2"
configure cfm group "erpsDn1" add rmep 15
configure cfm group "erpsDn2" add rmep 12
configure erps subring1 cfm port east add group erpsDn2
configure erps subring1 cfm port west add group erpsDn1
```

Sub-ring Configuration

First, configure a main ring on the Interconnected node:

```
create erps main-ring1
configure erps main-ring1 add ring-ports east 5
configure erps main-ring1 add ring-ports west 6
configure erps ring1 add control “cv1”
```

Next, configure a sub-ring on the interconnected node:

```
create erps sub-ring1
configure erps sub-ring1 add ring-ports east 10
configure erps sub-ring1 add control “subv1”
configure erps main-ring1 add sub-ring sub-ring1
enable erps main-ring1
enable erps sub-ring1
```

Virtual Channel for Sub-ring

```
configure vlan subv1 add port 5 6 tagged
configure main-ring1 add protected vlan subv1
```

No Virtual Channel for Sub-ring

```
configure erps subring1 subring-mode no-virtualChannel
```

Debugging ERPS

1. Check the output of `show erps` *`ring`* **`statistics`** to see if any error/dropped counters are incrementing.
   a. If they are, check the state of the ring ports and trace these links to the neighbor node to see the state of the links.

      The output of `show log` after turning on the filters for *ERPS* should provide more information on what is happening on the switch.
2. Check the output of `show erps` and `show erps` *`ring`* to see if the node state is as expected.

   In steady state, the node should be in “Idle” and the failed state ring should be in “Protected” state.

ERPS Feature Limitations

The following are *ERPS* feature limitations:

- Backup failover and checkpointing for both v1 and v2 are not available in the current release.
- In platforms that do not have hardware OAM (operations and management), the optimum CFM interval recommended is one second for link monitoring, which will give rise to approximately three-second overhead in convergence times.

  Note
  For optimum performance and convergence, it is recommended to use fiber cables.

- Other than the basic *EAPS* interoperability stated above, all other EAPS related interoperability is not supported.
- There is no interoperability with *STP (Spanning Tree Protocol)* in the current release.

ESRP

ESRP Overview on page 618
Configuring ESRP on page 629
Operation with Other ExtremeXOS Features on page 633
Advanced ESRP Features on page 636
Display ESRP Information on page 642
ESRP Configuration Examples on page 643

Extreme Standby Router Protocol™ (ESRP) allows multiple switches to provide redundant routing services to users. This chapter discusses how to configure, operate and display the ESRP feature. It also provides detailed configuration examples.

ESRP Overview

The *ESRP (Extreme Standby Router Protocol)*™, like the Virtual Router Redundancy Protocol (VRRP), allows multiple switches to provide redundant routing services to users.

ESRP is used to eliminate the single point of failure associated with manually configuring a default gateway address on each host in a network. Without using ESRP, if the configured default gateway fails, you must reconfigure each host on the network to use a different router as the default gateway. ESRP provides a redundant path for the hosts. Using ESRP, if the default gateway fails, the backup router assumes forwarding responsibilities.

Note
Support for ESRP operation over IPv6 networks was added in ExtremeXOS release 12.6.

In addition to providing Layer 3 routing redundancy for IP and IPX, ESRP also provides Layer 2 redundancy features for fast failure recovery and to provide for dual-homed system design. In some instances, depending on network system design, ESRP can provide better resiliency than using Spanning Tree Protocol (STP) or Virtual Router Redundancy Protocol (*VRRP (Virtual Router Redundancy Protocol)*). You can use Layer 3 and Layer 2 redundancy features in combination or independently. ESRP is available only on Extreme Networks switches. An example ESRP topology is shown in Figure 63.

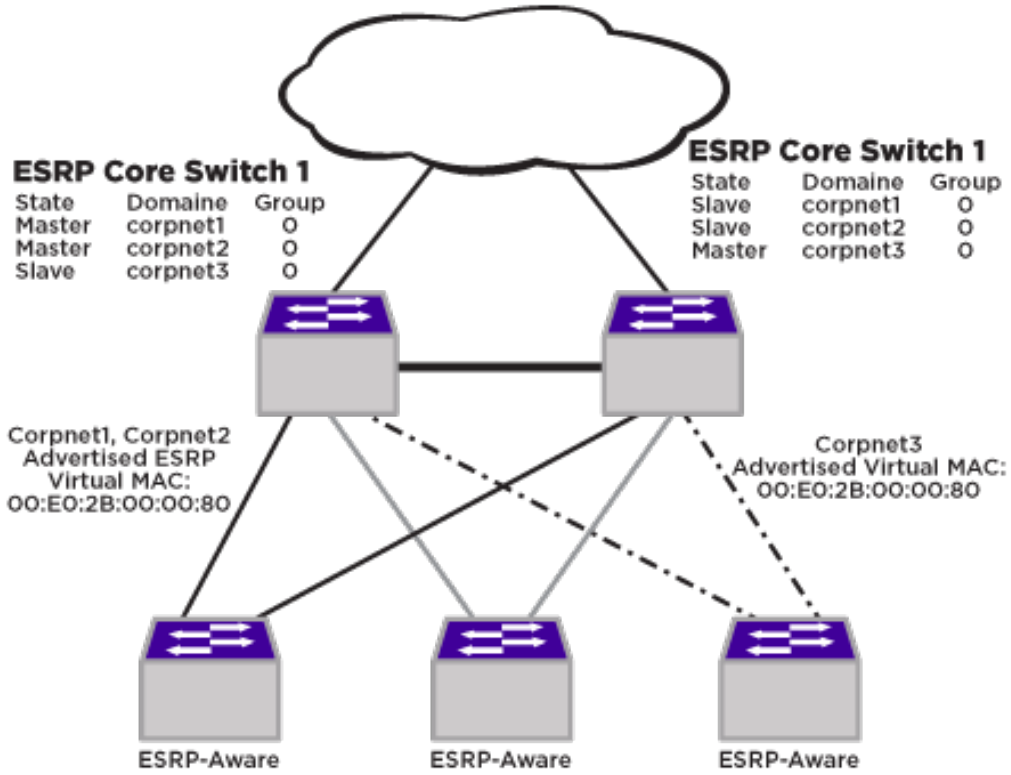


Figure 63: Example of a Basic ESRP Topology

In this example, ESRP Core Switch #1 and ESRP Core Switch #2 are both configured with three ESRP domains. Each domain represents a separate ESRP instance and supports a unique set of VLANs. Each domain is configured to use one master *VLAN (Virtual LAN)* and can support additional member VLANs. The switches exchange keep-alive packets for each VLAN independently.

Unless groups are configured, each ESRP domain supports two routers, one operating in the master state, and one operating in the slave state. Within an ESRP domain, any ESRP router can become the master, but only one ESRP router can be master at a time. Only the master can actively provide Layer 3 routing and/or Layer 2 switching for each VLAN. The master handles the forwarding, ARP requests, NDP messages, and routing for a particular VLAN. The slave router stands by, ready to take over if the master is no longer available.

Each switch in an ESRP topology has its own unique IP address (or IPX NetID) and a MAC address, which are required for basic IP connectivity. For each ESRP domain, there is a shared virtual IP address or IPX NetID and a MAC address, which are used for network client communications. The virtual IP address or IPX NetID is configured on all ESRP routers in a domain, and it is configured as the default gateway address on network clients in that domain. If the master ESRP router becomes unavailable, the backup ESRP router takes over using the same virtual IP address or IPX NetID.

The topology in Figure 63 shows that one switch serves as the master for the Corpnet1 and Corpnet2 domains, and the other switch serves as master for the Corpnet3 domain. This topology demonstrates the load sharing capability of ESRP. If one switch served as master for all ESRP domains, all traffic would be routed through that master, and the slave switch would be idle. Dividing the ESRP domain mastership between routers allows domain clients access to more bandwidth and reduces the likelihood of exceeding the capacity of a single master router.

You can use ESRP to achieve edge-level or aggregation-level redundancy.

Deploying ESRP in this area of the network allows you to simplify your network design, which is important in designing a stable network. ESRP also works well in meshed networks where Layer 2 loop protection and Layer 3 redundancy are simultaneously required.

Note
For complete information about platform support for ESRP, see the Switch Engine v33.6.1 Licensing Guide document.

ESRP Master Election

The system determines the *ESRP* master switch (providing Layer 3 routing and/or Layer 2 switching services for a *VLAN*) using the following default factors:

- Stickiness—The switch with the higher sticky value has higher priority. When an ESRP domain claims master, its sticky value is set to 1 (available only in extended mode).
- Active ports—The switch that has the greatest number of active ports takes highest precedence.
- Tracking information—Various types of tracking are used to determine if the switch performing the master ESRP function has connectivity to the outside world. ExtremeXOS software supports the following types of tracking:
  - VLAN—Tracks any active port connectivity to one designated VLAN. An ESRP domain can track one VLAN, and the tracked VLAN should not be a member of any other ESRP domain in the system.
  - IP unicast route table entry—Tracks specific learned routes from the IP route table.
  - Ping—Tracks *ICMP (Internet Control Message Protocol)* ping connectivity to specified devices.
  - Environment (health checks)—Tracks the environment of the switch, including power supplies.

  If any of the configured tracking mechanisms fail, the master ESRP switch relinquishes status as master, and remains in slave mode for as long as the tracking mechanism continues to fail.
- ESRP priority—This is a user-defined field. The range of the priority value is 0 to 255; a higher number has higher priority, except for 255. The default priority setting is 0. A priority setting of 255 makes an ESRP switch a standby switch that remains in slave

mode until you change the priority setting. We recommend this setting for system maintenance. A switch with a priority setting of 255 will never become the master.

- System MAC address—The switch with the higher MAC address has higher priority.
- Active port weight—The switch that has the highest port weight takes precedence. The bandwidth of the port automatically determines the port weight (available only in extended mode).

You can configure the precedence order of the factors used by the system to determine the master ESRP switch. For more information about configuring the ESRP election metrics, see ESRP Election Algorithms on page 623.

Master Switch Behavior

If a switch is master, it actively provides Layer 3 routing services to other *VLAN*s, and Layer 2 switching between all the ports of that VLAN.

Additionally, the switch exchanges *ESRP* packets with other switches that are in slave mode.

Pre-Master Switch Behavior

A pre-master switch is ready to transition to master, but is going through possible loop detection before changing to the master state.

Upon entering the pre-master state, the switch sends *ESRP* packets to other switches on that same *VLAN*. If the switch finds itself superior to its neighbor, and successfully executes loop detection techniques, the switch transitions to master. This temporary state avoids the possibility of having simultaneous masters.

Slave Switch Behavior

If a switch is in slave mode, it exchanges *ESRP* packets with other switches on that same *VLAN*.

When a switch is in slave mode, it does not perform Layer 3 routing or Layer 2 switching services for the VLAN. From a Layer 3 routing protocol perspective (for example, *RIP (Routing Information Protocol)* or *OSPF (Open Shortest Path First)*), when in slave mode for the VLAN, the switch marks the router interface associated with that VLAN as down. From a Layer 2 switching perspective, no forwarding occurs between the member ports of the VLAN; this prevents loops and maintains redundancy.

If you configure the switch to use the optional ESRP Host Attach configuration, the switch continues Layer 2 forwarding to the master. For more information, see ESRP Host Attach on page 640.

Neutral Switch Behavior

The neutral state is the initial state entered into by the switch.

In a neutral state, the switch waits for *ESRP* to initialize and run. A neutral switch does not participate in ESRP elections. If the switch leaves the neutral state, it enters the slave state.

Electing the PrimarySwitch

A new primary can be elected. This is done in one of the following ways:

- A communicated parameter change
- Loss of communication between primary and secondary

If a parameter determines the primary changes (for example, link loss or priority change), the election of the new primary typically occurs within one second. A parameter change triggers a handshake between the routers. As long as both routers agree upon the state transition, new primary election is immediate.

If a switch in secondary mode loses its connection with the primary, a new election occurs (using the same precedence order indicated ESRP Master Election on page 620 or using a configured precedence order described in ESRP Election Algorithms on page 623). The new election typically takes place in three times the defined timer cycle (8 seconds by default).

Before the switch transitions to the primary state, it enters a temporary pre-primary state. While in the pre-primary state, the switch sends *ESRP* PDUs until the pre-primary state timeout expires. Depending upon the election algorithm, the switch may then enter the primary or secondary state. Traffic is unaffected by the pre-primary state because the primary continues to operate normally. The pre-primary state avoids the possibility of having simultaneous primaries.

To configure the pre-primary state timeout, use the following command:

```
configure esrp esrpDomain timer premaster seconds
```

Caution

Configure the pre-primary state timeout only with guidance from Extreme Networks support. Misconfiguration can severely degrade the performance of ESRP and your switch.

ESRP Failover Time

ESRP Failover time is largely determined by the following factors:

- ESRP hello timer setting.
- ESRP neighbor timer setting.
- The routing protocol being used for interrouter connectivity if Layer 3 redundancy is used; *OSPF* failover time is faster than *RIP* failover time.

The failover time associated with the ESRP protocol depends on the timer setting and the nature of the failure. The default hello timer setting is two seconds; the range is 2-1024 seconds. The default neighbor timer setting is eight seconds; the range is 3*hello to 1024 seconds. The failover time depends on the type of event that caused ESRP to failover. In most cases, a non-hardware failover is less than one second, and a hardware failover is eight seconds.

If routing is configured, the failover of the particular routing protocol (such as RIP V1, RIP V2, or OSPF) is added to the failover time associated with ESRP.

If you use OSPF, make your OSPF configuration passive. A passive configuration acts as a stub area and helps decrease the time it takes for recalculating the network. A passive configuration also maintains a stable OSPF core.

For more information about the ESRP timers and configuring the ESRP timers, see ESRP Configuration Overview on page 630.

ESRP Election Algorithms

You configure the switch to use one of 15 different election algorithms to select the *ESRP* master. ESRP uses the default election policy for extended mode. If you have an ESRP domain operating in standard mode, the domain ignores the sticky and weight algorithms.

To change the election algorithm, you must first disable the ESRP domain and then configure the new election algorithm. If you attempt to change the election algorithm without disabling the domain first, an error message appears.

- To disable the ESRP domain, use the following command:

  ```
  disable esrp {esrpDomain}
  ```

- To modify the election algorithm, use the following command:

  ```
  configure esrp esrpDomain add elrp-poll ports [ports | all]
  ```

  If you attempt to use an election algorithm not supported by the switch, an error message similar to the following appears:

  ```
  ERROR: Specified election-policy is not supported! Supported Policies:
  1. sticky > ports > weight > track > priority > mac 2. ports > track
  > priority 3. sticky > ports > track > priority 4. ports > track >
  priority > mac 5. sticky > ports > track > priority > mac 6. priority
  > mac 7. sticky > priority > mac 8. priority > ports > track > mac 9.
  sticky > priority > ports > track > mac 10. priority > track > ports
  > mac 11. sticky > priority > track > ports > mac 12. track > ports
  > priority 13. sticky > track > ports > priority 14. track > ports >
  priority > mac 15. sticky > track > ports > priority > mac
  ```

ESRP Election Algorithms

The following table describes the ESRP election algorithms. Each algorithm considers the election factors in a different order of precedence. The election algorithms that use sticky and weight are only available in extended mode.

Table 39: ESRP Election Algorithms

Election Algorithm	Description
ports > track > priority	Specifies that this ESRP domain should consider election factors in the following order: active ports, tracking information, ESRP priority.
ports > track > priority > mac	Specifies that this ESRP domain should consider election factors in the following order: active ports, tracking information, ESRP priority, MAC address. Note: This is the default election algorithm for standard mode.
priority > mac	Specifies that this ESRP domain should consider election factors in the following order: ESRP priority, MAC address.
priority > ports > track > mac	Specifies that this ESRP domain should consider election factors in the following order: ESRP priority, active ports, tracking information, MAC address.
priority > track > ports > mac	Specifies that this ESRP domain should consider election factors in the following order: ESRP priority, tracking information, active ports, MAC address.
sticky > ports > track > priority	Specifies that this ESRP domain should consider election factors in the following order: stickiness, active ports, tracking information, ESRP priority.
sticky > ports > track > priority > mac	Specifies that this ESRP domain should consider election factors in the following order: stickiness, active ports, tracking information, ESRP priority, MAC address.
sticky > ports > weight > track > priority > mac	Specifies that this ESRP domain should consider election factors in the following order: stickiness, active ports, port weight, tracking information, ESRP priority, MAC address. This is the default election algorithm for extended mode.
sticky > priority > ports > track > mac	Specifies that this ESRP domain should consider election factors in the following order: stickiness, ESRP priority, active ports, tracking information, MAC address.
sticky > priority > track > ports > mac	Specifies that this ESRP domain should consider election factors in the following order: stickiness, ESRP priority, tracking information, active ports, MAC address.

Table 39: ESRP Election Algorithms (continued)

Election Algorithm	Description
sticky > priority > mac	Specifies that this ESRP domain should consider election factors in the following order: stickiness, ESRP priority, MAC address.
sticky > track > ports > priority	Specifies that this ESRP domain should consider election factors in the following order: stickiness, tracking information, active ports, ESRP priority.
sticky > track > ports > priority > mac	Specifies that this ESRP domain should consider election factors in the following order: stickiness, tracking information, active ports, ESRP priority, MAC address.
track > ports > priority	Specifies that this ESRP domain should consider election factors in the following order: tracking information, active ports, ESRP priority.
track > ports > priority > mac	Specifies that this ESRP domain should consider election factors in the following order: tracking information, active ports, ESRP priority, MAC address.

Caution
All switches in the ESRP network must use the same election algorithm; otherwise, loss of connectivity, broadcast storms, or other unpredictable behavior may occur.

ESRP Domains

ESRP domains allow you to configure multiple *VLAN*s under the control of a single instance of the ESRP protocol. By grouping multiple VLANs under one ESRP domain, the ESRP protocol can scale to provide protection to large numbers of VLANs. All VLANs within an ESRP domain simultaneously share the same active and standby router and failover router, as long as one port of each member VLAN belongs to the domain master.

Depending on the election policy used, when a port in a member VLAN belongs to the domain master, the member VLAN ports are considered when determining the ESRP master. You can configure a maximum of 64 ESRP domains in a network.

If you disable an ESRP domain, the switch notifies its neighbor that the ESRP domain is going down, and the neighbor clears its neighbor table. If the master switch receives this information, it enters the neutral state to prevent a network loop. If the slave switch receives this information, it also enters the neutral state.

ESRP packets do not identify themselves to which domain they belong; you either configure a domain ID or the ESRP domain uses the 802.1Q tag (VLANid) of the master VLAN.

A domain ID in the packet clearly classifies the packet, associates a received ESRP PDU to a specific ESRP domain, and tells the receiving port where the packet came from.

Note
Active Ports Count (ports): Total number of active physical ports of master VLANs of the ESRP domain.

ESRP Groups

ExtremeXOS software supports running multiple instances of *ESRP* within the same *VLAN* or broadcast domain. This functionality is called an *ESRP group*. Although other uses exist, the most typical application for multiple ESRP groups is when two or more sets of ESRP switches are providing fast-failover protection within a subnet.

A maximum of seven distinct ESRP groups can be supported on a single ESRP switch, and a maximum of 32 ESRP groups can be defined within the same network broadcast domain. You can configure a maximum of 32 ESRP groups in a network.

For example, two ESRP switches provide Layer 2/Layer 3 connectivity and redundancy for the subnet, while another two ESRP switches provide Layer 2 connectivity and redundancy for a portion of the same subnet. Figure 64 shows ESRP groups.

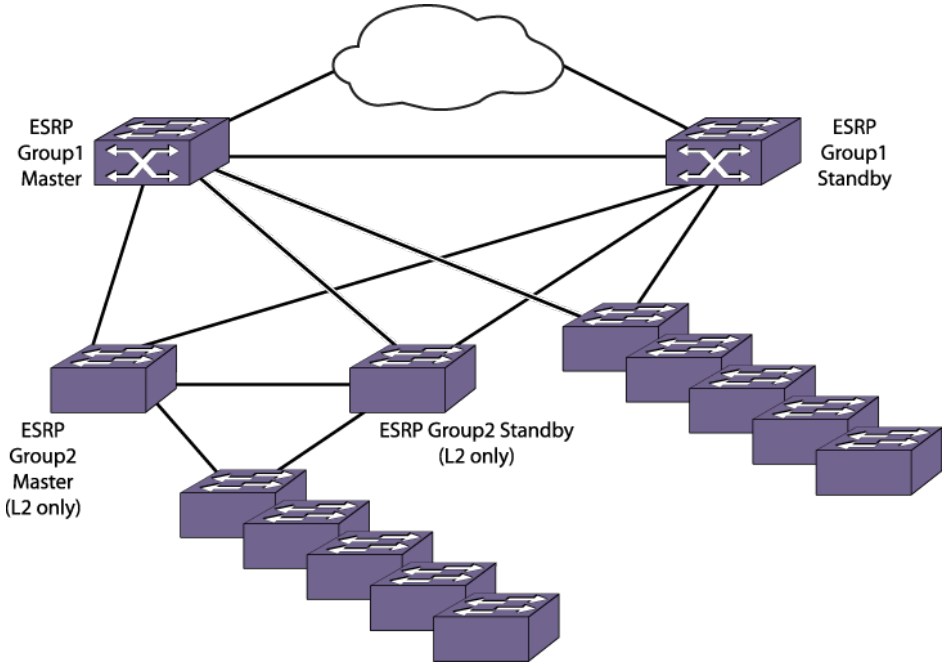


Figure 64: ESRP Groups

ESRP Extended Mode Features

The *ESRP* extended mode is enabled by default and provides the maximum ESRP feature set.

You can use ESRP extended mode only when all switches that participate in ESRP are running ExtremeXOS software.

The following list describes the ESRP extended mode features that are not available in standard mode:

- Handshaking

  In standard mode, events such as link flapping cause the ESRP master switch to generate a large number of packets and to increase processing time.

  To prevent this, extended mode supports handshaking, which occurs when a switch requests a state change, forces its neighbor to acknowledge the change, and the neighbor sends an acknowledgement to the requesting switch. For example, if a slave switch wants to become the master, it enters the pre-master state, notifies the neighbor switch, and forces the neighbor to acknowledge the change. The neighbor then sends an acknowledgement back to the slave switch. While the requesting switch waits for the acknowledgements, future updates are suppressed to make sure the neighbor does not act on incorrect data.
- Stickiness

  In standard mode, if an event causes the ESRP master switch to fail over to the slave, it becomes the new master. If another event occurs, the new master switch returns to the slave and you have experienced two network interruptions.

  To prevent this, extended mode supports the sticky election metric. The default election algorithm uses the sticky metric. For example, if an event causes the ESRP master switch to fail over to the slave, it becomes the new master and has a higher sticky value. If another event occurs, for example adding active ports to the slave, the new master does not fail back to the original master even if the slave has more active ports. After sticky is set on the master, regardless of changes to its neighbor's election algorithm, the new master retains its position. Sticky algorithms provide for fewer network interruptions than non-sticky algorithms. Sticky is set only on the master switch.
- Port weight

  In standard mode, the port count calculation does not take into account the available bandwidth of the ports. For example, a switch with a one Gigabit Ethernet uplink may be unable to become master because another switch has a load-shared group of four fast Ethernet links. The active port count calculation considers only the number of active ports, not the bandwidth of those ports.

  In extended mode, the active port count calculation considers the number of active ports and the port weight configuration considers the bandwidth of those ports. You enable port weight only on the load-shared master port.
- Domain ID

In standard mode, ESRP packets do not contain domain information; therefore, the only information about the packet comes from the receiving port.

The concept of domain ID is applicable only to extended mode. A domain ID in the packet clearly classifies the packet, associates a received ESRP PDU to a specific ESRP domain, and tells the receiving port where the packet came from. In extended mode, you must have a domain ID for each ESRP domain. Each switch participating in ESRP for a particular domain must have the same domain ID configured.

The ESRP domain ID is determined from one of the following user-configured parameters:

- ESRP domain number created with the command:

  `configure esrp esrpDomain` **`domain-id`** *`number`*
- 802.1Q tag (VLANid) of the tagged master *VLAN*
- Hello messages

  In standard mode, both the master switch and slave switch send periodic ESRP hello messages. This causes an increase in packet processing by both the master and slave.

  In extended mode, the master switch sends periodic ESRP hello messages. This reduces the amount of packet processing, increases the amount of available link bandwidth, and does not impact communicating state changes between switches.
- ESRP Automatic Toggle Feature

  ESRP includes an automatic toggle feature, which toggles to the same mode of operation as an ESRP neighbor. This action causes the switch to enter the neutral state and re-elect the ESRP master.

  Note
  The automatic toggle feature toggles the ESRP operational mode, not the configured mode. If the switch is configured for ESRP extended mode and the switch toggles to standard mode, the switch enters extended mode the next time the switch boots.

Linking ESRP Switches

When considering system design using *ESRP*, we recommend using a direct link.

Direct links between ESRP switches are useful under the following conditions:

- A direct link can provide a more direct routed path, if the ESRP switches are routing and supporting multiple *VLAN*s where the master/slave configuration is split such that one switch is master for some VLANs and a second switch is master for other VLANs. The direct link can contain a unique router-to-router VLAN/subnet, so that the most direct routed path between two VLANs with different master switches uses a direct link, instead of forwarding traffic through another set of connected routers.

- A direct link can be used as a highly reliable method to exchange ESRP hello messages, so that the possibility of having multiple masters for the same VLAN is lessened if all downstream Layer 2 switches fail.
- A direct link is necessary for the ESRP host attach option. The direct link is used to provide Layer 2 forwarding services through an ESRP slave switch.

Direct links may contain a router-to-router VLAN, along with other VLANs participating in an ESRP domain. If multiple VLANs are used on the direct links, use 802.1Q tagging. The direct links may be aggregated into a load-shared group, if desired. If multiple ESRP domains share a host port, each VLAN must be in a different ESRP group.

ESRP-Aware Switches

Extreme Networks switches that are not actively participating in *ESRP* but are connected on a network that has other Extreme Networks switches running ESRP are ESRP-aware. When ESRP-aware switches are attached to ESRP-enabled switches, the ESRP-aware switches reliably perform failover and failback scenarios in the prescribed recovery times.

If Extreme Networks switches running ESRP are connected to Layer 2 switches that are manufactured by third-party vendors, the failover times for traffic local to that segment may appear longer, depending on the application involved and the *FDB (forwarding database)* timer used by the other vendor's Layer 2 switch. ESRP can be used with Layer 2 switches from other vendors, but the recovery times vary.

The *VLAN*s associated with the ports connecting an ESRP-aware switch to an ESRP-enabled switch must be configured using an 802.1Q tag on the connecting port; or, if only a single VLAN is involved, as untagged using the protocol filter 'any.' ESRP does not function correctly if the ESRP-aware switch interconnection port is configured for a protocol-sensitive VLAN using untagged traffic. You can also use port restart in this scenario. For more information, see ESRP Port Restart on page 639.

The following sections provide information on managing ESRP-aware switches:

- Configuring ESRP-Aware Switches on page 632
- ESRP Configuration Examples on page 643

Configuring ESRP

Guidelines

To participate in *ESRP*, the following must be true:

- A *VLAN* can belong to only one ESRP domain.
- The IP address for the VLANs participating in an ESRP domain must be identical.
- For operation over IPv6, both an IPv6 and an IPv4 address must be present on the master VLAN for every participating router.

- All switches in the ESRP network must use the same election algorithm, otherwise loss of connectivity, broadcast storms, or other unpredictable behavior may occur.
- If you have an untagged master VLAN, you must specify an ESRP domain ID. The domain ID must be identical on all switches participating in ESRP for that particular domain.
- If you have a tagged master VLAN, ESRP uses the 802.1Q tag (VLANid) of the master VLAN for the ESRP domain ID. If you do not use the VLANid as the domain ID, you must specify a different domain ID. As previously described, the domain ID must be identical on all switches participating in ESRP for that particular domain.

Note

If you configure the *OSPF* routing protocol and ESRP, you must manually configure an OSPF router identifier. Be sure that you configure a unique OSPF router ID on each switch running ESRP. For more information, see OSPF

We recommend that all switches participating in ESRP run the same version of ExtremeXOS software.

ESRP Configuration Overview

The following procedure can be used to configure a simple *ESRP* topology:

1. Create and configure a *VLAN* that will become the master VLAN. (See VLANs on page 2213.)
2. As needed, create and configure the VLANs that will become the member VLANs. (See VLANs on page 2213.)
3. Create the ESRP domain as described in Creating and Deleting an ESRP Domain on page 631.
4. If your configuration requires an ESRP domain ID, configure it as described in .
5. Add the master VLAN to the ESRP domain as described in Adding and Deleting a Master VLAN on page 631.
6. If your configuration requires member VLANs, add the member VLANs to the ESRP domain as described in Adding and Deleting a Member VLAN on page 632.
7. Enable ESRP for the specified ESRP domain as described in Enabling and Disabling an ESRP Domain on page 632.

You can also configure other ESRP domain parameters, including ESRP:

- Timers as described in the Switch Engine v33.6.1 Command References .
- Election algorithms as described in ESRP Election Algorithms on page 623.
- Tracking as described in ESRP Tracking on page 636.
- Port restart as described in ESRP Port Restart on page 639.
- Host attach as described in ESRP Host Attach on page 640.
- Groups as described in ESRP Groups on page 626.

For more detailed information about all of the commands used to create, configure, enable, and disable an ESRP domain, refer to the Switch Engine v33.6.1 Command References .

Creating and Deleting an ESRP Domain

You can specify a unique *ESRP* domain name to identify each ESRP domain in your network.

- To create an ESRP domain, use the following command:

  `create esrp esrp_domain {type [vpls-redundancy | standard]}`

  The *esrpDomain* parameter is a character string of up to 32 characters that identifies the ESRP domain to be created.

  Note

  If you use the same name across categories (for example, *STPD (Spanning Tree Domain)* and ESRP names) we recommend that you specify the appropriate keyword as well as the actual name. If you do not specify the keyword, the switch may display an error message.

- To delete an ESRP domain, use the following command:

  `delete esrp esrpDomain`

Configuring the ESRP Domain ID

If you choose not use the 802.1Q tag (VLANid) of the master *VLAN*, or you have an untagged master VLAN, you must create a domain ID before you can enable the *ESRP* domain. For more information about ESRP domains and the ESRP domain ID, see ESRP Domains on page 625.

- To configure an ESRP domain ID, use the following command:

  `configure esrp esrpDomain domain-id number`

  The *number* parameter specifies the number of the domain ID. The user-configured ID range is 4096 through 65,535.

Adding and Deleting a Master VLAN

The master *VLAN* is the VLAN on the *ESRP* domain that exchanges ESRP PDUs and data between a pair of ESRP-enabled devices. You must configure one master VLAN for each ESRP domain, and a master VLAN can belong to only one ESRP domain.

- To add a master VLAN to an ESRP domain, use the following command:

  `configure esrp esrpDomain add master vlan_name`

  The *esrpDomain* parameter specifies the name of the ESRP domain, and the *vlan_name* parameter specifies the name of the master VLAN.

- To delete a master VLAN, you must first disable the ESRP domain before removing the master VLAN using the `disable esrp {esrpDomain}` command.
- To delete a master VLAN from an ESRP domain, use the following command:.

  `configure esrp esrpDomain delete master vlan_name`

Adding and Deleting a Member VLAN

The member *VLAN* can belong to only one *ESRP* domain, and you configure zero or more member VLANs for each ESRP domain.The state of the ESRP device determines whether the member VLAN is in the forwarding or blocking state.

- To add a member VLAN to an ESRP domain. use the following command:

  configure esrp esrpDomain add member vlan_name

  The *esrpDomain* parameter specifies the name of the ESRP domain, and the *vlan_name* parameter specifies the name of the member VLAN.
- To delete a member VLAN from an ESRP domain, use the following command:

  configure esrp esrpDomain delete member vlan_name

Enabling and Disabling an ESRP Domain

- To enable a specific *ESRP* domain, use the following command:

  enable esrp *esrpDomain*
- To disable a specific ESRP domain, use the following command:

  disable esrp {*esrpDomain*}

Configuring ESRP-Aware Switches

For an Extreme Networks switch to be ESRP-aware, you must create an ESRP domain on the aware switch, add a master *VLAN* to that ESRP domain, add a member VLAN to that ESRP domain if configured, and configure a domain ID if necessary.

To participate as an ESRP-aware switch, the following must be true:

- The ESRP domain name must identical on all switches (ESRP-enabled and ESRP-aware) participating in ESRP for that particular domain.
- The master VLAN name and IP address must be identical on all switches (ESRP-enabled and ESRP-aware) participating in ESRP for that particular domain.
- If configured, the member VLAN name and IP address must be identical on all switches (ESRP-enabled and ESRP-aware) participating in ESRP for that particular domain.
- The domain ID must be identical on all switches (ESRP-enabled or ESRP-aware) participating in ESRP for that particular domain.
- If you have an untagged master VLAN, you must specify an ESRP domain ID.
- If you have a tagged master VLAN, ESRP uses the 802.1Q tag (VLANid) of the master VLAN for the ESRP domain ID. If you do not use the VLANid as the domain ID, you must specify a different domain ID.

Note

Before you begin, make a note of the ESRP domain parameters on the ESRP-enabled switch. That way you can easily refer to your notes while creating the ESRP domain on the ESRP-aware switch.

To configure an ESRP-aware switch:

1. Create an ESRP domain using the command:

   ```
   create esrp esrp_domain {type [vpls-redundancy | standard]}
   ```

   For complete information about software licensing for this feature, see the Switch Engine v33.6.1 Licensing Guide document.
2. Add a master VLAN to your ESRP domain using the command:

   ```
   configure esrp esrpDomain add master vlan_name
   ```
3. If configured, add the member VLANs to your ESRP domain using the command:

   ```
   configure esrp esrpDomain add member vlan_name
   ```
4. If necessary, configure a domain ID for the ESRP domain using the command:

   ```
   configure esrp esrpDomain domain-id number
   ```

Operation with Other ExtremeXOS Features

ESRP and IP Multinetting

When configuring *ESRP* and IP multinetting on the same switch, the same set of IP addresses must be configured for all involved VLANs.

ESRP and STP

A switch running *ESRP* should not simultaneously participate in *STP (Spanning Tree Protocol)* for the same *VLAN*(s). Other switches in the VLAN being protected by ESRP may run STP; the switch running ESRP forwards, but does not filter, STP BPDUs. Therefore, you can combine ESRP and STP on a network and a VLAN, but you must do so on separate devices. You should be careful to maintain ESRP connectivity between ESRP master and slave switches when you design a network that uses ESRP and STP.

ESRP and VRRP

Do not configure *ESRP* and *VRRP* on the same *VLAN* or port. This configuration is not allowed or supported.

ESRP Groups and Host Attach

ESRP domains that share ESRP HA ports must be members of different ESRP groups.

Port Configurations and ESRP

The following ports cannot be part of a *VLAN* that participates in an *ESRP* domain:

- A mirroring target port
- A software-controlled redundant port
- A Netlogin port

In addition, the following ESRP ports cannot be a mirroring, software-controlled redundant port, or Netlogin port:

- Host Attach port
- Don't-count port (This port has a port weight of 0.)
- Restart port

Using ELRP with ESRP

Extreme Loop Recovery Protocol (ELRP) is a feature of ExtremeXOS software that allows you to prevent, detect, and recover from Layer 2 loops in the network.

You can use ELRP with other protocols, including *ESRP*.

Note

When used on a VPLS service *VLAN*, ELRP does not detect loops involving the VPLS pseudowires.

For more information about standalone ELRP, see Using ELRP to Perform Loop Tests on page 2118.

With ELRP, each switch, except for the sender, treats the ELRP protocol data unit (PDU) as a Layer 2 multicast packet. The sender uses the source and destination MAC addresses to identify the packet it sends and receives. When the sender receives its original packet back, that triggers loop detection and prevention. After a loop is detected, the loop recovery agent is notified of the event and takes the necessary actions to recover from the loop. ELRP operates only on the sending switch; therefore, ELRP operates transparently across the network.

How a loop recovers is dependent upon the protocol that uses the loop detection services provided by ELRP.

If you are using ELRP in an ESRP environment, ESRP may recover by transitioning the ESRP domain from master to slave. The following sections describe how ESRP uses ELRP to recover from a loop and the switch behavior:

- Using ELRP with ESRP to Recover Loops on page 634
- Configuring ELRP on page 635
- Displaying ELRP Information on page 636

Using ELRP with ESRP to Recover Loops

ELRP sends loop-detect packets to notify *ESRP* about loops in the network.

In an ESRP environment, when the current master goes down, one of the slaves becomes the master and continues to forward Layer 2 and Layer 3 traffic for the ESRP domain. If a situation occurs when a slave incorrectly concludes that the master is down, the slave incorrectly assumes the role of master. This introduces more than one master on the ESRP domain which causes temporary loops and disruption in the network.

ELRP on an ESRP Pre-Master Switch

A pre-master switch is an ESRP switch that is ready to transition to master but is going through possible loop detection.

A pre-master periodically sends out ELRP loop-detect packets (ELRP PDUs) for a specified number of times and waits to make sure that none of the sent ELRP PDUs are received. Transition to master occurs only after this additional check is completed. If any of the ELRP PDUs are received, the switch transitions from pre-master to slave state. You configure pre-master ELRP loop detection on a per ESRP domain basis.

ELRP on an ESRP Master Switch

A master switch is an ESRP switch that sends ELRP PDUs on its ESRP domain ports.

If the master switch receives an ELRP PDU that it sent, the master transitions to the slave. While in the slave state, the switch transitions to the pre-master rate and periodically checks for loops before transitioning to the master. The pre-master process is described in ELRP on an ESRP Pre-Master Switch on page 635. You configure the master ELRP loop detection on a per ESRP domain basis.

Configuring ELRP

By default, ELRP is disabled. The following sections describe the commands used to configure ELRP for use with *ESRP*:

- Configuring Pre-Master Polling on page 635
- Configuring Master Polling on page 635
- Configuring Ports on page 636

Note

When used on a virtual private LAN service (VPLS) *VLAN*, ELRP does not detect loops involving the VPLS pseudowires.

Configuring Pre-Master Polling

If you enable the use of ELRP by *ESRP* in the pre-master state, ESRP requests ELRP packets sent to ensure that there is no loop in the network before changing to the master state. If no packets are received, there is no loop in the network. By default, the use of ELRP by ESRP in the pre-master state is disabled.

- To enable the use of ELRP by ESRP in the pre-master state on a per-ESRP domain basis, and to configure how often and how many ELRP PDUs are sent in the pre-master state, use the following command:
  `configure esrp esrpDomain elrp-premaster-poll enable {count count | interval interval}`
- To disable the use of ELRP by ESRP in the pre-master state, use the following command:
  `configure esrp esrpDomain elrp-premaster-poll disable`

Configuring Master Polling

If you enable the use of ELRP by *ESRP* in the master state, ESRP requests that ELRP packets are periodically sent to ensure that there is no loop in the network while ESRP

is in the master state. By default, the use of ELRP by ESRP in the master state is disabled.

- To enable the use of ELRP by ESRP in the master state on a per-ESRP domain basis, and to configure how often the master checks for loops in the network, use the following command:

  `configure esrp esrpDomain elrp-master-poll enable {interval interval}`

- To disable the use of ELRP by ESRP in the master state, use the following command:

  `configure esrp esrpDomain elrp-master-poll disable`

Configuring Ports

You can configure one or more ports of an *ESRP* domain where ELRP packet transmission is requested by ESRP. This allows the ports in your network that might experience loops, such as ports that connect to the master, slave, or ESRP-aware switches, to receive ELRP packets. You do not need to send ELRP packets to host ports.

By default, all ports of the ESRP domain have ELRP transmission enabled on the ports.

If you change your network configuration, and a port no longer connects to a master, slave, or ESRP-aware switch, you can disable ELRP transmission on that port.

- To disable ELRP transmission, use the following command:

  `configure esrp esrpDomain delete elrp-poll ports [ports | all]`

- To enable ELRP transmission on a port, use the following command:

  `configure esrp esrpDomain add elrp-poll ports [ports | all]`

Displaying ELRP Information

To display summary ELRP information, use the following command:

`show elrp`

In addition to displaying the enabled/disabled state of ELRP, the command displays the total number of:

- Clients registered with ELRP
- ELRP packets transmitted
- ELRP packets received

For more information about the output associated with the `show elrp` command, see the Switch Engine v33.6.1 Command References .

Advanced ESRP Features

ESRP Tracking

Tracking information is used to track various forms of connectivity from the *ESRP* switch to the outside world.

ESRP Environment Tracking

You can configure *ESRP* to track hardware status. If a power supply fails, if the switch is overheating, or if a non-fully loaded power supply is detected, the priority for the ESRP domain will change to the failover settings.

Note
ExtremeXOS software determines the maximum available power required for the switch by calculating the number of power supplies and the power required by the installed modules. Enabling environmental tracking on the switch without enough power budget causes tracking to fail. In this case, the tracking failure occurs by design.

To configure the failover priority for an ESRP domain:

1. Set the failover priority using the following command:

   `configure esrp esrpDomain add track-environment failover priority`

2. Assign the priority flag precedence over the active ports count using the following command:

   `configure esrp esrpDomain election-policy [ports > track > priority | ports > track > priority > mac | priority > mac | priority > ports > track > mac | priority > track > ports > mac | sticky > ports > track > priority | sticky > ports > track > priority > mac | sticky > ports > weight > track > priority > mac | sticky > priority > mac | sticky > priority > ports > track > mac | sticky > priority > track > ports > mac | sticky > track > ports > priority | sticky > track > ports > priority > mac | track > ports > priority | track > ports > priority > mac]`

Because the priority of both ESRP domains are set to the same value, ESRP will use the active ports count to determine the master ESRP domain.

ESRP VLAN Tracking

You can configure an *ESRP* domain to track port connectivity to a specified *VLAN* as criteria for ESRP failover. The number of VLAN active ports are tracked. If the switch is no longer connected to the specified VLAN, the switch automatically relinquishes master status and remains in slave mode. You can track a maximum of one VLAN.

- To add a tracked VLAN, use the following command:

  `configure esrp esrpDomain add track-vlan vlan_name`

- To delete a tracked VLAN, use the following command:

  `configure esrp esrpDomain delete track-vlan vlan_name`

ESRP Unicast Route Table Tracking

You can configure *ESRP* to track specified IPv4 routes in the route table as criteria for ESRP failover. If all of the configured routes are not available within the route table, the

switch automatically relinquishes master status and remains in slave mode. You can track a maximum of eight routes per route table.

Note
ESRP route tracking is not supported for IPv6 destinations..

- To add a tracked route, use the following command:
  `configure esrp esrpDomain add track-iproute ipaddress/masklength`
- To delete a tracked route, use the following command:
  `configure esrp esrpDomain delete track-iproute ipaddress/masklength`

ESRP Ping Tracking

You can configure *ESRP* to track connectivity using a simple ping to any IPv4 device. This may represent the default route of the switch, or any device meaningful to network connectivity of the master ESRP switch. The switch automatically relinquishes master status and remains in slave mode if a ping keepalive fails. You can configure a maximum of eight ping tracks.

Note
ESRP ping tracking is not supported for IPv6 destinations. The ESRP ping tracking option cannot be configured to ping an IP address within an ESRP *VLAN* subnet. It should be configured on some other normal VLAN across the router boundary.

- To configure ping tracking, use the following command:
  `configure esrp esrpDomain` **`add track-ping`** *`ipaddress`* {**`frequency`** *`seconds`*} {**`miss`** *`misses`*} {**`success`** *`successes`*}

  The **`frequency`** option specifies the number of seconds between ping requests.

  The **`miss`** option specifies the number of consecutive ping failures that initiate failover to an ESRP slave.

  The **`success`** option specifies the number consecutive ping successes required for tracking success.
- To disable ping tracking, use the following command:
  `configure esrp esrpDomain delete track-ping ipaddress`

Displaying ESRP Tracking Information

You can view the status of *ESRP* tracking on a per domain basis. The information displayed includes the type of tracking used by the ESRP domain and how you configured the tracking option.

To view the status of tracked devices, use the following command:

`show esrp name`

ESRP Tracking Example

Figure 65 is an example of *ESRP* tracking.

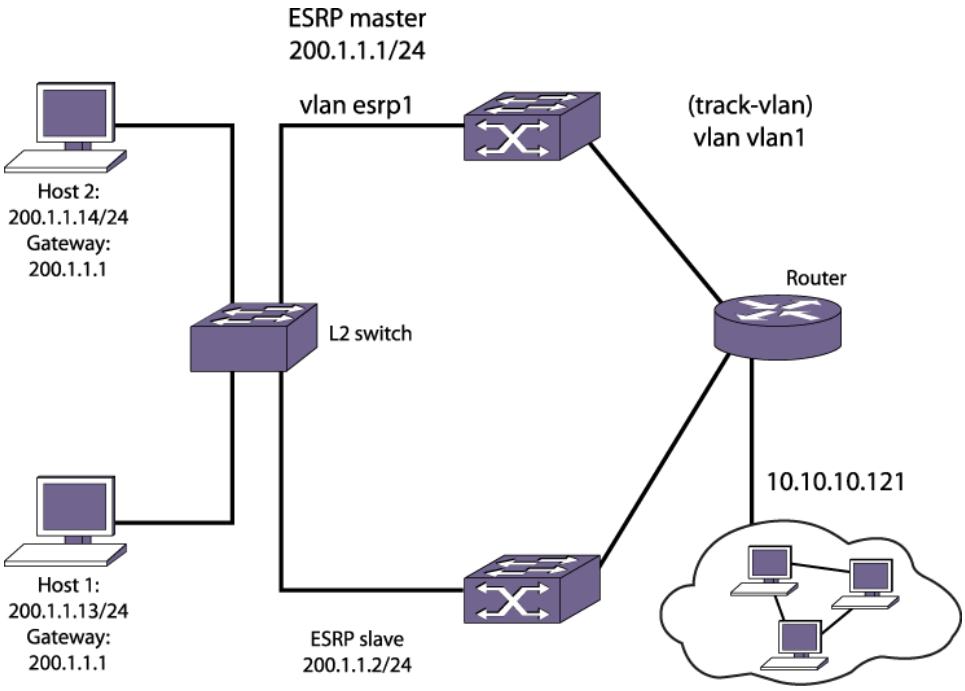


Figure 65: ESRP Tracking

- To configure *VLAN* tracking, use the following command:

  ```
  configure esrp esrp1 add track-vlan vlan1
  ```

  Using the tracking mechanism, if VLAN1 fails, the ESRP master realizes that there is no path to the upstream router via the master switch and implements an ESRP failover to the slave switch.
- To configure route table tracking, use the following command:

  ```
  configure esrp esrp1 add track-iproute 10.10.10.0/24
  ```

  The IPv4 route specified in this command must exist in the IP routing table. When the route is no longer available, the switch implements an ESRP failover to the slave switch.
- To configure ping tracking, use the following command:

  ```
  configure esrp esrp1 add track-ping 10.10.10.121 frequency 2 miss 2
  ```

  The specified IPv4 address is tracked. If the fail rate is exceeded, the switch implements an ESRP failover to the slave switch.

ESRP Port Restart

You can configure *ESRP* to restart ports in the ESRP master domain when the downstream switch is from a third-party vendor. This action takes down and restarts the port link to clear and refresh the downstream ARP table.

- To configure port restart, use the following command:

  ```
  configure esrp ports ports restart
  ```

- To disable port restart, use the following command:

  ```
  configure esrp ports ports no-restart
  ```

  If a switch becomes a slave, ESRP takes down (disconnects) the physical links of member ports that have port restart enabled. The disconnection of these ports causes downstream devices to remove the ports from their *FDB* tables. This feature allows you to use ESRP in networks that include equipment from other vendors. After 2 seconds, the ports re-establish connection with the ESRP switch.
- To remove a port from the restart configuration, delete the port from the *VLAN* and re-add it.

ESRP Host Attach

ESRP host attach (HA) is an optional ESRP configuration that allows you to connect active hosts directly to an ESRP master or slave switch.

Normally, the Layer 2 redundancy and loop prevention capabilities of ESRP do not allow packet forwarding from the slave ESRP switch. ESRP HA allows configured ports that do not represent loops to the network to continue Layer 2 operation independent of their ESRP status.

ESRP HA is designed for redundancy for dual-homed server connections. HA allows the network to continue Layer 2 forwarding regardless of the ESRP status. Do not use ESRP HA to interconnect devices on the slave ESRP switch instead of connecting directly to the ESRP master switch.

The ESRP HA option is useful if you are using dual-homed network interface cards (NICs) for server farms, as shown in the following figure. The ESRP HA option is also useful where an unblocked Layer 2 environment is necessary to allow high-availability security.

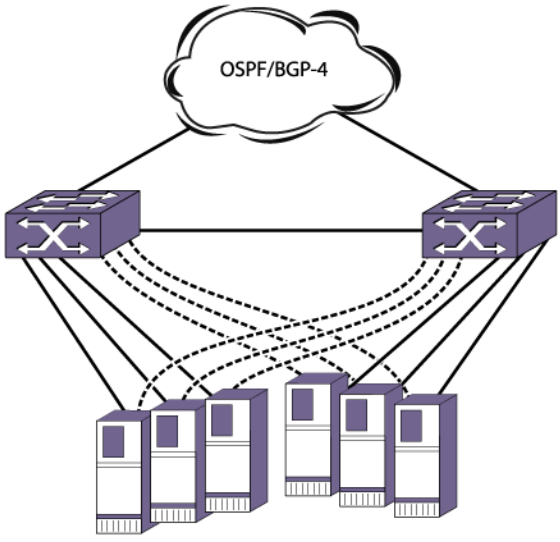


Figure 66: ESRP Host Attach

If you use load sharing with the ESRP HA feature, configure the load-sharing group first and then enable HA on the group.

Other applications allow lower-cost redundant routing configurations because hosts can be directly attached to the switch involved with ESRP. HA also requires at least one link between the master and the slave ESRP switch for carrying traffic and to exchange ESRP hello packets.

Note

Do not use the ESRP HA feature with the following protocols: *STP* or *VRRP*. A broadcast storm may occur.

To configure a port to be a host port, use the following command: `configure esrp ports ports mode [host | normal]`

ESRP Port Weight and Don't Count

In an *ESRP* domain, the switch automatically calculates the port weight based on the bandwidth of the port. ESRP uses the port weight to determine the master ESRP switch.

For load-shared ports, configure the master port in the load-share group with the port weight. A load-shared port has an aggregate weight of all of its member ports. If you add or delete a load-shared port (or trunk), the master load-shared port weight is updated.

If you do not want to count host ports and normal ports as active, configure the ESRP port weight on those ports. Their weight becomes 0 and that allows the port to be part of the *VLAN*, but if a link failure occurs, it will not trigger a reconvergence. With this configuration, ESRP experiences fewer state changes due to frequent client activities like rebooting and unplugging laptops. This port is known as a don't-count port.

- To configure the port weight on either a host attach port or a normal port, use the following command:

  `configure esrp ports ports weight [auto | port-weight]`

Selective Forwarding

An *ESRP*-aware switch floods ESRP PDUs from all ports in an ESRP-aware *VLAN*. This flooding creates unnecessary network traffic because some ports forward ESRP PDUs to switches that are not running the same ESRP groups. You can select the ports that are appropriate for forwarding ESRP PDUs by configuring selective forwarding on an ESRP-aware VLAN and thus reduce this excess traffic. Configuring selective forwarding creates a port list of only those ports that forward to the ESRP groups that

are associated with an ESRP-aware VLAN. This ESRP-aware port list is then used for forwarding ESRP PDUs.

Note

We recommend keeping the default settings unless you have considerable knowledge and experience with ESRP.

- To configure selective forwarding, use the following command:

  `configure esrp domain aware add selective-forward-ports portlist {group group number}`

  When an ESRP-aware switch receives an ESRP PDU on a domain, the software looks up the group to which the PDU belongs. If the group is found, the ESRP-aware switch processes the PDU then and forwards it according to the group's specified aware selective forwarding port list. If no selective forwarding port list is configured, the switch forwards the PDU from all of the ports of the domain's master VLAN. If the group is not found, the PDU is forwarded on all ports.

  When a user adds one or more ports to the ESRP-aware port list (for example, 5:1 and 6:2) that are not part of the master VLAN, the following message appears:

  ```
  Warning: Port 5:1, 6:2 not currently a member of master vlan
  ```

  The ports will still be added to the ESRP-aware port list; however, PDUs will not be forwarded out of those ports until they are added to the master VLAN.
- To disable selective forwarding, use the following command:

  `configure esrp domain aware delete selective-forward-ports all| portlist {group group number}`

Display Selective Forwarding Information

To display all selective forwarding information for a given domain, use the following command:

```
show esrp domain aware {selective-forward-ports | statistics}
```

Display ESRP Information

- To view *ESRP* information, use the command `show esrp`

  Output from this command includes:

  - The operational state of an ESRP domain and the state of its neighbor.
  - ESRP port configurations.
- To view more detailed information about an ESRP domain on an ESRP enabled switch or an ESRP aware switch, use the following command and specify the domain name:

  `show esrp { {name} | {type [vpls-redundancy | standard]} }`

- To view ESRP counter information for a specific domain, use the following command:

  ```
  show esrp {name} counters
  ```

- To view ESRP-aware information for a specific domain (including the group number, MAC address for the master, and the age of information), use the following command:

  ```
  show esrp domain aware {selective-forward-ports | statistics}
  ```

ESRP Configuration Examples

Single Domain Using Layer 2 and Layer 3 Redundancy

The example shown in Figure 67 uses four Extreme Networks devices as edge switches that perform Layer 2 switching for *VLAN* Sales.

The edge switches are dual-homed two switches that are configured to support *ESRP* domain esrp1. The ESRP switches perform Layer 2 switching and Layer 3 routing between the edge switches and the outside world. Each ESRP switch has the VLAN Sales configured using the identical IP address. The ESRP switches then connect to the routed enterprise normally, using the desired routing protocol (for example, *RIP* or *OSPF*).

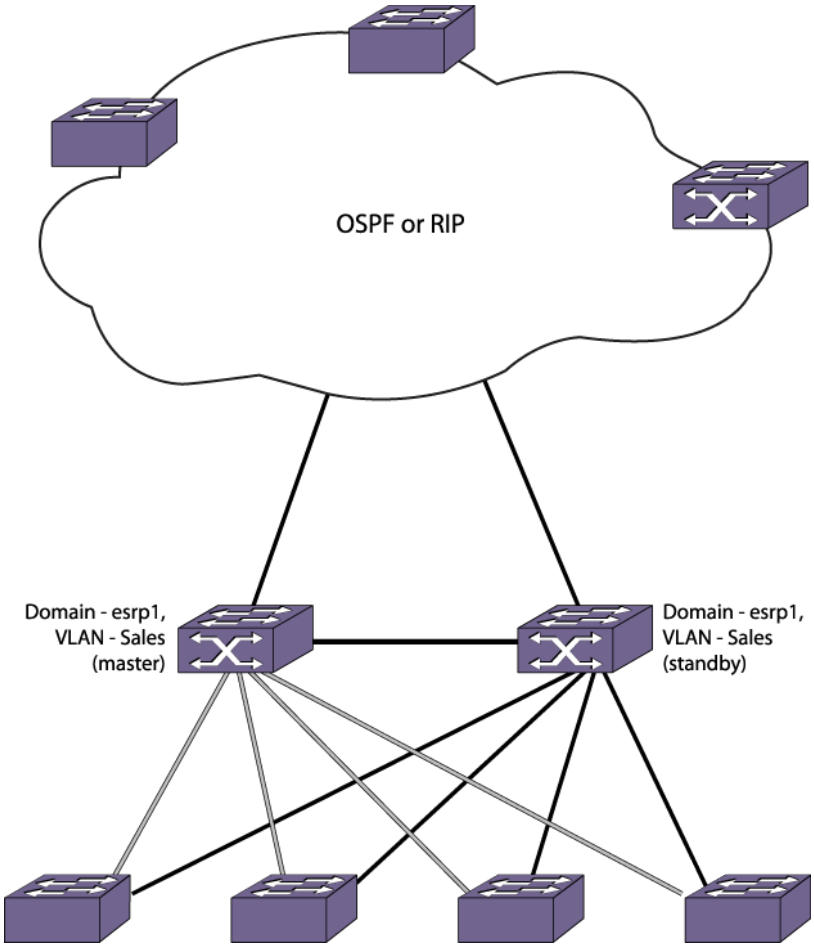


Figure 67: Single ESRP Domain Using Layer 2 and Layer 3 Redundancy

In this example, the ESRP master performs both Layer 2 switching and Layer 3 routing services for VLAN Sales. To prevent bridging loops in the VLAN, the ESRP slave performs no switching or routing for VLAN Sales while the ESRP master is operating.

There are four paths between each ESRP switch and the edge switches for VLAN Sales. All the paths are used to send ESRP packets, allowing for four redundant paths for communication. The edge switches, being ESRP-aware, allow traffic within the VLAN to failover quickly because these edge switches sense when a master/slave transition occurs and flush *FDB* entries associated with the uplinks to the ESRP-enabled switches.

This example assumes the following:

- ESRP election algorithm used is the default for standard mode (ports > track > priority > mac).
- The inter-router backbone is running OSPF, with other routed VLANs already properly configured. Similar commands would be used to configure a switch on a network running RIP.
- Ports added to the VLAN have already been removed from VLAN default.

- The same IP address is specified for all VLANs participating in ESRP.
- The master is determined by the programmed MAC address of the switch because the number of active links for the VLAN and the priority are identical for both switches.

The following commands are used to configure both ESRP switches:

```
create vlan sales
configure vlan sales add ports 1:1-1:4
configure vlan sales ipaddress 10.1.2.3/24
enable ipforwarding
create esrp esrp1
configure esrp esrp1 domain-id 4096
configure esrp esrp1 add master sales
enable esrp esrp1
configure ospf add vlan sales area 0.0.0.0 passive
configure ospf routerid 5.5.5.5
enable ospf
```

Multiple Domains Using Layer 2 and Layer 3 Redundancy

The example shown in Figure 68 illustrates an *ESRP* configuration that has multiple domains using Layer 2 and Layer 3 redundancy.

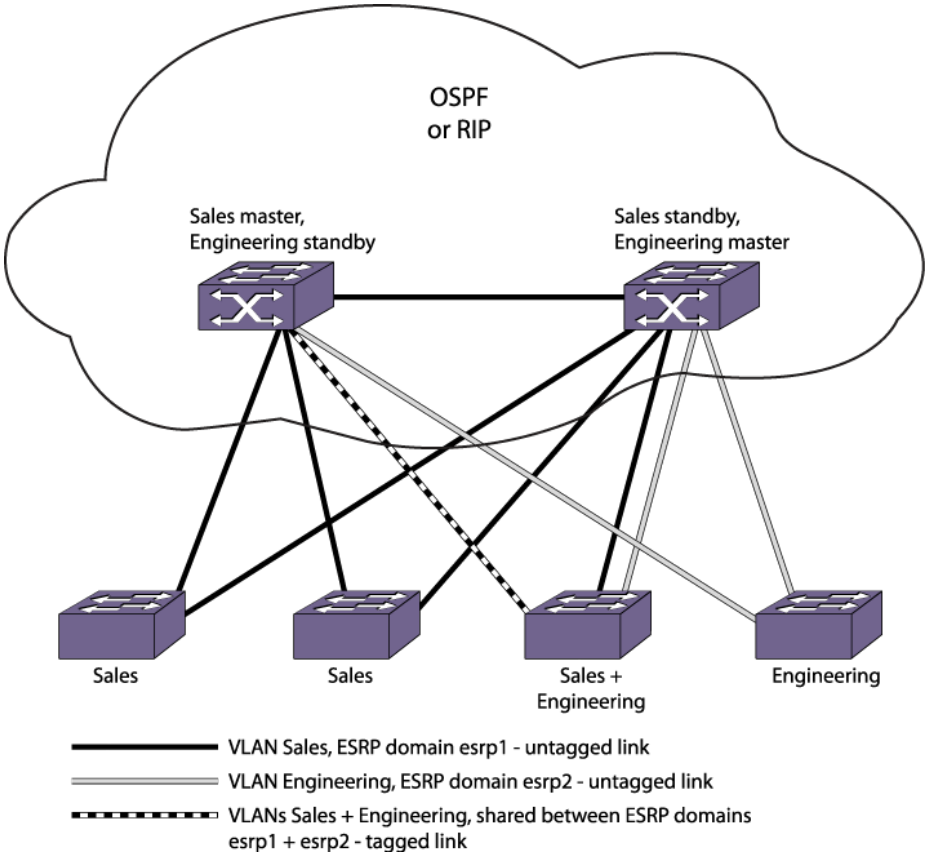


Figure 68: Multiple ESRP Domains Using Layer 2 and Layer 3 Redundancy

This example builds on the previous example. It has the following features:

- An additional *VLAN*, Engineering, is added that uses Layer 2 redundancy.
- VLAN Sales uses three active links to each ESRP switch.
- VLAN Engineering has two active links to each ESRP switch.
- One of the edge devices carries traffic for both VLANs.
- The link between the third edge device and the first ESRP switch uses 802.1Q tagging to carry traffic from both VLANs on one link. The ESRP switch counts the link active for each VLAN.
- The second ESRP switch has a separate physical port for each VLAN connected to the third edge switch.

In this example, the ESRP switches are configured such that VLAN Sales normally uses the first ESRP switch and VLAN Engineering normally uses the second ESRP switch. This is accomplished by manipulating the ESRP priority setting for each VLAN for the particular ESRP switch.

Configuration commands for the first ESRP switch are as follows:

```
create vlan sales
configure vlan sales tag 10
configure vlan sales add ports 1:1-1:2
configure vlan sales add ports 1:3, 1:5 tagged
configure vlan sales ipaddress 10.1.2.3/24
create vlan engineering
configure vlan engineering tag 20
configure vlan engineering add ports 1:4
configure vlan engineering add ports 1:3, 1:5 tagged
configure vlan engineering ipaddress 10.4.5.6/24
create esrp esrp1
configure esrp esrp1 domain-id 4096
configure esrp esrp1 add master sales
configure esrp esrp1 priority 5
enable esrp esrp1
create esrp esrp2
configure esrp esrp2 domain-id 4097
configure esrp esrp2 add master engineering
enable esrp esrp2
```

Configuration commands for the second ESRP switch are as follows:

```
create vlan sales
configure vlan sales tag 10
configure vlan sales add ports 1:1-1:3
configure vlan sales ipaddress 10.1.2.3/24
configure vlan sales add ports 1:5 tagged
create vlan engineering
configure vlan engineering tag 20
configure vlan engineering add ports 1:4, 2:1
configure vlan engineering ipaddress 10.4.5.6/24
configure vlan engineering add ports 1:5 tagged
create esrp esrp1
configure esrp esrp1 domain-id 4096
configure esrp 1 add master sales
enable esrp esrp1
create esrp esrp2
configure esrp esrp2 domain-id 4097
configure esrp esrp2 add master engineering
```

```
configure esrp esrp2 priority 5
enable esrp esrp2
```

ESRP Over IPv6 Configuration Example

The example shown in the following figure illustrates an *ESRP* configuration that can operate over IPv6.

Note

To support operation over IPv6, the master VLANs configured in this example require both an IPv4 and an IPv6 address. ESRP route tracking and ESRP ping tracking are not supported for IPv6 addresses.

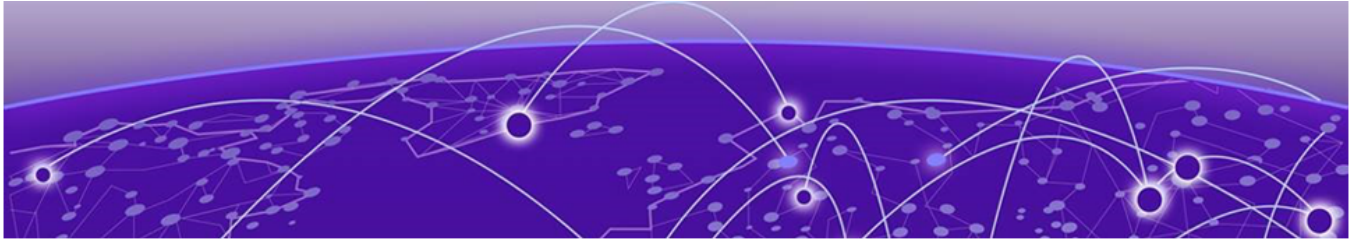
Configuration commands for the first ESRP switch are as follows:

```
create vlan sales
configure vlan sales tag 10
configure vlan sales add ports 1:1-1:2
configure vlan sales add ports 1:3, 1:5 tagged
configure vlan sales ipaddress 10.1.2.3/24
configure vlan sales ipaddress 2001:db8:36::1/48
create vlan engineering
configure vlan engineering tag 20
configure vlan engineering add ports 1:4
configure vlan engineering add ports 1:3, 1:5 tagged
configure vlan engineering ipaddress 10.4.5.6/24
configure vlan engineering ipaddress 2001:db8:36::2/48
create esrp esrp1
configure esrp esrp1 domain-id 4096
configure esrp esrp1 add master sales
configure esrp esrp1 priority 5
enable esrp esrp1
create esrp esrp2
configure esrp esrp2 domain-id 4097
configure esrp esrp2 add master engineering
enable esrp esrp2
```

Configuration commands for the second ESRP switch are as follows:

```
create vlan sales
configure vlan sales tag 10
configure vlan sales add ports 1:1-1:3
configure vlan sales ipaddress 10.1.2.3/24
configure vlan sales ipaddress 2001:db8:36::1/48
configure vlan sales add ports 1:5 tagged
create vlan engineering
configure vlan engineering tag 20
configure vlan engineering add ports 1:4, 2:1
configure vlan engineering ipaddress 10.4.5.6/24
configure vlan engineering ipaddress 2001:db8:36::2/48
configure vlan engineering add ports 1:5 tagged
create esrp esrp1
configure esrp esrp1 domain-id 4096
configure esrp 1 add master sales
enable esrp esrp1
create esrp esrp2
configure esrp esrp2 domain-id 4097
configure esrp esrp2 add master engineering
```

```
configure esrp esrp2 priority 5
enable esrp esrp2
```



Ethernet Virtual Private Network (EVPN)

EVPN Overview on page 649
Configuring EVPN Overview on page 651
Displaying EVPN Information on page 654
EVPN Configuration Examples on page 656

This chapter explains the ExtremeXOS® implementation of *BGP (Border Gateway Protocol)*-based Ethernet VPNs (EVPNs) as described in *RFC 7209*, *RFC 7432*, and *RFC 8365*.

This chapter provides information on configuring and managing EVPN, displaying EVPN information, and configuration examples.

EVPN Overview

BGP was standardized in *RFC 7432* and *RFC 8365* to carry Layer-2 information for virtualized networks. Ethernet virtual private network (EVPN) was initially targeted for *MPLS (Multiprotocol Label Switching)* and WAN, but later adopted as a *VXLAN (Virtual Extensible LAN)* control plane protocol. ExtremeXOS supports EVPN control plane for VXLAN. At a high level, BGP supports the following constructs to distribute information for any virtualized network:

- **Identify the network address family with AFI/SAFI**—For VXLAN, AFI of 25 (L2VPN) and a SAFI (EVPN) of 70 is used.
- **Within the address family, identify the type of route being advertised**—Different route types are defined. For supported route types in ExtremeXOS, see Table 40 on page 651.
- **Identify which device originated the route and virtual-network the route belongs to**—This is the role of the route distinguisher (RD) and route target (RT).

In ExtremeXOS, RD is automatically derived based on configured local-endpoint IP address and the VLAN ID of the tenant VLAN. The VLAN ID of the tenant VLAN is also used as an EVPN instance (EVI) index. An EVI is automatically created when a virtual network configuration is completed and BGP is enabled. A complete virtual-network consists of a tenant VLAN, VNI, and local endpoint IP address. RTs are automatically calculated for each EVI using the method specified in *Section 5.1.2.1* of *RFC-8365*. Because this method uses the local AS number as part of the calculation, each router

calculates different RTs in an EBGP environment because the local AS number is different.

Note

RTs are derived automatically under the following conditions:

- When iBGP is used and the AS number is less than 65,535 (2 byte ASNUM).
- BGP Auto-peering is used.

EVPN devices perform dynamic learning on the access side of a VXLAN network. The learned MAC and ARP entries are exported to BGP as EVPN Type 2 routes. BGP advertises the Type 2 routes to all neighbors with L2VPN-EVPN capability. A VTEP with matching RT configured or auto-derived processes these Type 2 routes and creates static non-permanent FDB, ARP tunnel entries. Traffic to the tunnel entries are VXLAN encapsulated and forwarded.

EVPN Supported Platforms

All ExtremeSwitching Universal switches.

EVPN requires a Premier license. For more information about licensing, see Switch Engine v33.6.1 Licensing Guide .

EVPN Limitations

The following are limitations for EVPN:

- EVPN functionality is not supported between switches running ExtremeXOS 30.2 and switches running earlier ExtremeXOS versions. The earlier versions rely on auto-creation of IBGP peers, which is disabled functionality in ExtremeXOS 30.2. However, the proprietary AFI are supported and can be used to establish tunnels to RTEPs so that native VXLAN functionality using data plane learning functions is supported.
- A maximum of 4,000 EVI instances are supported.
- IPv6 Type 2 routes are not supported.
- Stacking is not supported.
- Configuring VMANs as VXLAN tenant VLANs is not supported.
- Multi-hop BFD is not supported.
- Peer-group configuration for L2VPN-EVPN address family is not supported.
- If silent hosts are expected, static ARP/FDB should be created on tenant VLANs for these hosts. To configure static ARP entries it is necessary to configure IP address on tenant VLANs.
- For Type 5 routes, the following are not supported:
  - Switching through a VXLAN tunnel to a remote L3 Anycast gateway.
  - Default VRs.

EVPN Supported Route Types

The following table shows the supported EVPN route types in ExtremeXOS.

Table 40: Supported EVPN Route Types in ExtremeXOS

Route Type	Route type Name	Usage	ExtremeXOS Support
1	Ethernet Auto Discovery per ESI and per EVI	Per ES- fast convergence, split horizon,per EVI—aliasing, backup path	No (MLAG used)
2	Mac/IP route	Reachability to MAC/IP.	Yes
3	Inclusive multicast ethernet tag route (PMSI tunnel)	VNI membership for broadcast and multicast replication	Yes
4	Ethernet Segment route	DF election, multi-homed ES auto-discovery	No
5	IP prefix route	Advertise IP prefixes	Yes, except on Default VR.
6	Selective multicast ethernet tag route	IGMP membership to EVPN	No

Configuring EVPN Overview

This section explains how to configure EVPN.

Recommended Configuration

The following is a recommended configuration for EVPN:

- ARP suppression should be enabled on tenant VLANs.
- Configure and apply IP Anycast on leafs (see IP and MAC Anycast on page 2228, and configure IP address on tenant VLAN and enable IP forwarding.
- IP route sharing should be enabled.
- BGP multipath should be enabled.

Configuring EVPN

To configure EVPN:

1. Configure underlay L3 network (see BGP and OSPF).
2. Configure tenant VLANs and virtual-networks (see VXLAN on page 2305).
3. Enable L2VPN-EVPN capability (see EVPN Commands on page 653).
4. If manual route targets (RTs) are required (see Applying Manual Route Targets to EVPN Instances on page 652):
   a. Create EVPN instances.

b. Configure EVPN instances.

5. If eBGP is used, set next-hop-unchanged for L2VPN-EVPN routes (see EVPN Commands on page 653).

Note
BGP Auto-peering (see BGP Auto-peering on page 484) automates most of the above steps. If you are using BGP Auto-peering, only Step 2 and configuring LTEP IP are required

Applying Manual Route Targets to EVPN Instances

Manual route targets (RTs) can be applied to EVPN instances (EVI) using BGP commands. This is required to ensure RTs match between EBGP peering routers.

To associate BGP configuration with an automatically generated EVI:

1. Create a named EVI (see EVPN Commands on page 653).
2. Configure a VNI for the named EVI that matches the VNI of the corresponding virtual-network (see VXLAN on page 2305).
3. Configure the desired RTs for the named EVI (see EVPN Commands on page 653).

For example:

Virtual Network Configuration

```
configure virtual-network local-endpoint ipaddress 1.0.0.25 vr "VR-Default"
create virtual-network "vnet1" flooding standard
configure virtual-network "vnet1" VXLAN vni 678
configure virtual-network "vnet1" add vlan vlan48
```

BGP EVPN Instance Configuration

```
create bgp EVPN instance “inst1”
configure bgp EVPN instance “inst1” VXLAN vni 678
configure bgp EVPN instance “inst1” route-target both add 10:1
```

BGP EVPN Partial Route Target Matching

The import of BGP/EVPN routes is controlled by route target (RT) extended communities. A route is imported if it contains at least one RT that matches a locally configured import RT on the receiving router. The import and export RTs are automatically generated using the *RFC-8365* algorithm, which includes the local BGP autonomous system (AS) as part of the calculated value.

In an external BGP (EBGP) peering session, since the routers reside in different ASs, the auto-generated RTs do not match, and routes are not imported. Currently, you can manually configure RTs on an EVPN instance to allow matching. For Auto-peering BGP sessions, as well as Type 5 EVPN routes, a common AS number in the private AS range is used in the auto-calculation to ensure matching. This feature provides support for “ignore-as” or also called “partial RT matching” functionality, which when enabled ignores the AS number portion of the RT when attempting to find a matching value. This reduces user configuration, since you do not have to use the CLI to create an EVPN

instance with RTs, and also allows for easier interoperability with third-party devices in cases where ExtremeXOS auto-generates using the private AS.

For command information, see EVPN Commands on page 653.

EVPN Commands

Sending/Receiving Routes

To enable the capability to send and receive EVPN routes, use the following command and select the l2vpn-evpn option:

```
enable bgp neighbor [all |remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]
```

To disable the capability to send and receive EVPN routes, use the following command and select the l2vpn-evpn:

```
disable bgp neighbor [all | remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]
```

Creating/Deleting EVPN Instances

To create an EVPN instance, use the following command:

```
create bgp evpn instance evpn_instance_name
```

The EVPN instance becomes active if the configured VNI matches the configured VNI of a virtual-network.

To delete an EVPN instance, use the following command:

```
delete bgp evpn instance evpn_instance_name
```

Configuring EVPN Instances

To add or delete a virtual extensible local area network (VXLAN) virtual network identifier (VNI) to an EVPN instance, use the following command:

```
configure bgp evpn instance evpn_instance_name vxlan vni [vni_value | none]
```

To configure route targets for an EVPN instance, use the following command:

```
configure bgp evpn instance evpn_instance_name route-target {import | export | both} [add | delete] route_target
```

To configure route distinguishers for an EVPN instance, use the following command:

```
configure bgp evpn instance evpn_instance_name rd [rd_value | auto]
```

Enabling/disabling Next-Hop Unchanged Behavior

To enable overriding the specification behavior with respect to the next-hop of routes advertised to EBGP peers, use the following command:

```
enable bgp {neighbor [remoteaddr | all]} {{address-family} l2vpn-evpn}
next-hop-unchanged
```

To disable overriding the specification behavior with respect to the next-hop of routes advertised to EBGP peers, use the following command:

```
disable bgp {neighbor [remoteaddr | all]} {{address-family} l2vpn-evpn}
next-hop-unchanged
```

EVPN Type 5 Routes

To configure a Layer 3 virtual network identifier (VNI) and bind it to a VPN VRF, use the following command:

```
configure bgp evpn l3vni [vni_value | none] vr vr_name
```

To export direct, static, and BGP routes from a VRF into BGP, running on the specified VR, as EVPN routes to be advertised by BGP as Type 5 routes, use the following command:

```
enable bgp export [static | direct] {address-family address_family}
l2vpn-evpn {vr vr_name}
```

To stop exporting direct, static, and BGP routes from a VRF into BGP, running on the specified VR, as EVPN routes to be advertised by BGP as Type 5 routes, use the following command:

```
disable bgp export [static | direct] {address-family address_family}
l2vpn-evpn {vr vr_name}
```

BGP EVPN Partial Route Target Matching

```
configure bgp evpn ignore-as [on | off]
```

Displaying EVPN Information

Showing BGP Information

To see general BGP neighbor information, including EVPN parameters, use the following command:

```
show bgp {neighbor} remoteaddr {address-family [ipv4-unicast | ipv4-
multicast |ipv6-unicast | ipv6-multicast | ipv4-vxlan | {l2vpn-evpn
[inclusive-multicast | mac-ip | auto-discovery | esi | ip-prefix]}]}
[accepted-routes | received-routes | rejected-routes | transmitted-
routes] {detail} [all | as-path path-expression | community [no-
advertise | no-export | no-export-subconfed | number community_number
```

```
| autonomous-system-id : bgp-community] | network [any/netMaskLen |
networkPrefixFilter] {exact}]
```

To see BGP route information, use the following command using the l2vpn-evpn option:

```
show bgp routes {address-family [ipv4-unicast | ipv4-multicast |ipv6-
unicast | ipv6-multicast | ipv4-vxlan | {l2vpn-evpn [inclusive-multicast
| mac-ip | auto-discovery | esi | ip-prefix]}]} {detail} [ipv4-vxlan |
all | as-path path-expression | community [no-advertise | no-export |
no-export-subconfed | number community_number | autonomous-system-idbgp-
community] | network [any/netMaskLen | networkPrefixFilter] {exact}] {vr
vr_name}
```

Showing EVPN Information

To show the EVPN global configuration, use the following command:

```
show bgp evpn
```

To show information about the EVPN instance table, use the following command:

```
show bgp evpn evi {evi-index evi_index} {vni vni}
```

To show the current set of MAC addresses configured in EVPN, use the following command:

```
show bgp evpn mac {mac-address mac_address}
```

To show the IPv4 entries from the EVPN MAC/IP table, use the following command:

```
show bgp evpn ipv4 {evi-index evi_index} {ip-address ip_address}
```

To show the IPv6 entries from the EVPN MAC/IP table, use the following command:

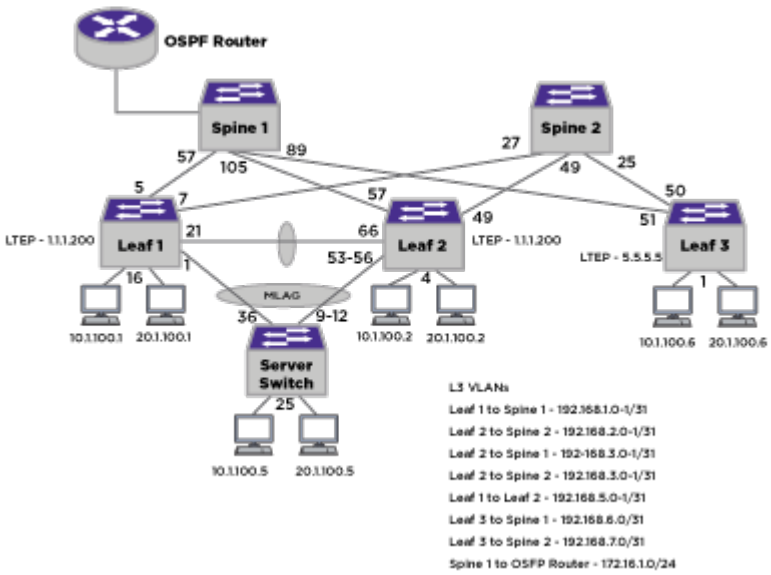
```
show bgp evpn ipv6 {evi-index evi_index} {ip-address ip_address}
```

To show the L3 VNI entries from the EVPN MAC/IP table, use the following command:

```
show bgp evpn l3vni {vr vr_name}}
```

EVPN Configuration Examples

EVPN with iBGP Configuration Example



Configuration Notes

iBGP auto-RT capability is used.

Note
If 4 byte AS numbers are used, RTs should be manually created.

- MLAG peers share the same LTEP IP, but use different BGP router IDs.
- OSPF adjacency between Leafs and Spines, between MLAG peers. PTP link type is used.
- OSPF routes are exported to BGP on Spine 1.
- On MLAG peers OSPF router IDs should be different (manually configured).
- Static LACP MAC is configured on MLAG peers.
- BGP maximum paths is set to 2 for this topology and IP route sharing is enabled.
- ARP suppression is enabled on tenant VLANs.
- Spines are configured as RR.

Note

- On SLX and other platforms “retain route-targets” should be enabled on spine/transit L3 nodes.ExtremeXOS by default retains all RTs and does not have a command to tun it off.
- ExtremeXOS does not support multi-hop BFD currently—under certain conditions convergence might take longer time.

Leaf 1 Configuration

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-36
configure vr VR-Default add ports 1-36
configure vlan default delete ports 1,4-5,7,16,19,21
enable jumbo-frame ports all
create vlan "isc"
configure vlan isc tag 4000
create vlan "loop"
enable loopback-mode vlan loop
create vlan "mlagvtep"
enable loopback-mode vlan mlagvtep
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "trunk1_2"
configure vlan trunk1_2 tag 4001
create vlan "trunk1_3"
create vlan "trunk1_4"
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 1 auto off speed 10000 duplex full
configure ports 2 auto off speed 10000 duplex full
configure ports 3 auto off speed 10000 duplex full
configure ports 4 auto off speed 10000 duplex full
configure ports 5 auto off speed 10000 duplex full
configure ports 6 auto off speed 10000 duplex full
configure ports 7 auto off speed 10000 duplex full
configure ports 8 auto off speed 10000 duplex full
configure ports 9 auto off speed 10000 duplex full
configure ports 10 auto off speed 10000 duplex full
configure ports 11 auto off speed 10000 duplex full
configure ports 12 auto off speed 10000 duplex full
configure ports 13 auto off speed 10000 duplex full
configure ports 14 auto off speed 10000 duplex full
configure ports 15 auto off speed 10000 duplex full
configure ports 16 auto off speed 10000 duplex full
configure ports 17 auto off speed 10000 duplex full
configure ports 18 auto off speed 10000 duplex full
configure ports 19 auto off speed 10000 duplex full
configure ports 20 auto off speed 10000 duplex full
configure ports 21 auto off speed 10000 duplex full
configure ports 22 auto off speed 10000 duplex full
configure ports 23 auto off speed 10000 duplex full
configure ports 24 auto off speed 10000 duplex full
configure ports 26 auto off speed 10000 duplex full
configure ports 27 auto off speed 10000 duplex full
configure ports 28 auto off speed 10000 duplex full
enable sharing 1 grouping 1 algorithm address-based L2 lacp
configure vlan Default add ports 2-3,6,8-15,17-18,20,22-36 untagged
configure vlan isc add ports 21 tagged
configure vlan tenant add ports 1,16,21 tagged
configure vlan trunk1_2 add ports 21 tagged
configure vlan trunk1_3 add ports 5 untagged
configure vlan trunk1_4 add ports 7 untagged
configure vlan untagtenant add ports 1,16,21 untagged
configure vlan isc ipaddress 172.16.1.0 255.255.255.254
configure vlan loop ipaddress 1.1.1.1 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.27 255.255.255.0
```

```
configure vlan mlagvtep ipaddress 1.1.1.200 255.255.255.255
enable ipforwarding vlan mlagvtep
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan trunk1_2 ipaddress 192.168.5.0 255.255.255.254
enable ipforwarding vlan trunk1_2
configure vlan trunk1_3 ipaddress 192.168.1.0 255.255.255.254
enable ipforwarding vlan trunk1_3
configure vlan trunk1_4 ipaddress 192.168.2.0 255.255.255.254
enable ipforwarding vlan trunk1_4
configure vlan untagtenant ipaddress 20.1.100.100 255.255.255.0
enable ipforwarding vlan untagtenant

#
# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"

#
# Module vsm configuration.
#
create mlag peer "leaf2"
configure mlag peer "leaf2" ipaddress 172.16.1.1 vr VR-Default
configure mlag peer "leaf2" lacp-mac 00:00:de:ad:be:ef
enable mlag port 1 peer "leaf2" id 1

#
# Module ospf configuration.
#
configure ospf routerid 1.1.1.1
enable ospf
configure ospf add vlan loop area 0.0.0.0
configure ospf vlan loop priority 0
configure ospf add vlan mlagvtep area 0.0.0.0
configure ospf add vlan trunk1_2 area 0.0.0.0 link-type point-to-point
configure ospf add vlan trunk1_3 area 0.0.0.0 link-type point-to-point
configure ospf add vlan trunk1_4 area 0.0.0.0 link-type point-to-point

#
# Module bgp configuration.
#
configure bgp AS-number 10000
configure bgp routerid 1.1.1.1
configure bgp maximum-paths 2
configure bgp add network 1.1.1.1/32
configure bgp add network 1.1.1.200/32
create bgp neighbor 2.2.2.2 remote-AS-number 10000
configure bgp neighbor 2.2.2.2 source-interface ipaddress 1.1.1.1
enable bgp neighbor 2.2.2.2
create bgp neighbor 3.3.3.100 remote-AS-number 10000
configure bgp neighbor 3.3.3.100 source-interface ipaddress 1.1.1.1
enable bgp neighbor 3.3.3.100
create bgp neighbor 4.4.4.4 remote-AS-number 10000
configure bgp neighbor 4.4.4.4 source-interface ipaddress 1.1.1.1
enable bgp neighbor 4.4.4.4
enable bgp neighbor 2.2.2.2 capability l2vpn-EVPN
enable bgp neighbor 3.3.3.100 capability l2vpn-EVPN
enable bgp neighbor 4.4.4.4 capability l2vpn-EVPN
enable bgp export direct address-family ipv4-unicast
enable bgp

#
# Module otm configuration.
```

```
#
configure virtual-network local-endpoint ipaddress 1.1.1.200 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant
```

Leaf 2 Configuration

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-72
configure vr VR-Default add ports 1-72
configure vlan default delete ports 1-72
enable jumbo-frame ports all
create vlan "isc"
configure vlan isc tag 4000
create vlan "loop"
enable loopback-mode vlan loop
create vlan "loop2"
enable loopback-mode vlan loop2
create vlan "mlagvtep"
enable loopback-mode vlan mlagvtep
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "trunk1_2"
configure vlan trunk1_2 tag 4001
create vlan "trunk2_3"
create vlan "trunk2_4"
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 50 auto off speed 10000 duplex full
configure ports 51 auto off speed 10000 duplex full
configure ports 52 auto off speed 10000 duplex full
enable sharing 53 grouping 53-56 algorithm address-based L2 lacp
configure vlan isc add ports 66 tagged
configure vlan tenant add ports 4-5,53,66 tagged
configure vlan trunk1_2 add ports 66 tagged
configure vlan trunk2_3 add ports 57 untagged
configure vlan trunk2_4 add ports 49 untagged
configure vlan untagtenant add ports 4-5,53,66 untagged
configure vlan isc ipaddress 172.16.1.1 255.255.255.254
configure vlan loop ipaddress 2.2.2.2 255.255.255.255
enable ipforwarding vlan loop
configure vlan loop2 ipaddress 2.2.2.100 255.255.255.255
enable ipforwarding vlan loop2
configure vlan Mgmt ipaddress 10.127.16.19 255.255.255.0
configure vlan mlagvtep ipaddress 1.1.1.200 255.0.0.0
enable ipforwarding vlan mlagvtep
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan trunk1_2 ipaddress 192.168.5.1 255.255.255.254
enable ipforwarding vlan trunk1_2
configure vlan trunk2_3 ipaddress 192.168.3.0 255.255.255.254
enable ipforwarding vlan trunk2_3
configure vlan trunk2_4 ipaddress 192.168.4.0 255.255.255.254
enable ipforwarding vlan trunk2_4
configure vlan untagtenant ipaddress 20.1.100.100 255.255.255.0
enable ipforwarding vlan untagtenant
```

```
#
# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"


#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.1.1.200 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant


#
# Module bgp configuration.
#
configure bgp AS-number 10000
configure bgp routerid 2.2.2.2
configure bgp maximum-paths 2
configure bgp add network 1.1.1.200/32
configure bgp add network 2.2.2.2/32
create bgp neighbor 1.1.1.1 remote-AS-number 10000
configure bgp neighbor 1.1.1.1 source-interface ipaddress 2.2.2.2
enable bgp neighbor 1.1.1.1
create bgp neighbor 3.3.3.100 remote-AS-number 10000
configure bgp neighbor 3.3.3.100 source-interface ipaddress 2.2.2.2
enable bgp neighbor 3.3.3.100
create bgp neighbor 4.4.4.4 remote-AS-number 10000
configure bgp neighbor 4.4.4.4 source-interface ipaddress 2.2.2.2
enable bgp neighbor 4.4.4.4
enable bgp neighbor 1.1.1.1 capability l2vpn-EVPN
enable bgp neighbor 3.3.3.100 capability l2vpn-EVPN
enable bgp neighbor 4.4.4.4 capability l2vpn-EVPN
enable bgp export direct address-family ipv4-unicast
enable bgp
#
# Module ospf configuration.
#
configure ospf routerid 2.2.2.2
enable ospf
configure ospf add vlan loop area 0.0.0.0
configure ospf add vlan mlagvtep area 0.0.0.0
configure ospf add vlan trunk1_2 area 0.0.0.0 link-type point-to-point
configure ospf add vlan trunk2_3 area 0.0.0.0 link-type point-to-point
configure ospf add vlan trunk2_4 area 0.0.0.0 link-type point-to-point

#
# Module vrrp configuration.
#
create vrrp vlan tenant vrid 1
configure vrrp vlan tenant vrid 1 priority 255
create vrrp vlan untagtenant vrid 1
configure vrrp vlan untagtenant vrid 1 priority 255
configure vrrp vlan tenant vrid 1 add 10.1.100.100
configure vrrp vlan untagtenant vrid 1 add 20.1.100.100
enable vrrp vlan tenant vrid 1
enable vrrp vlan untagtenant vrid 1
```

```
#
# Module vsm configuration.
#
create mlag peer "leaf1"
configure mlag peer "leaf1" ipaddress 172.16.1.0 vr VR-Default
configure mlag peer "leaf1" lacp-mac 00:00:de:ad:be:ef
enable mlag port 53 peer "leaf1" id 1
```

Leaf 3

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-56
configure vr VR-Default add ports 1-56
configure vlan default delete ports 1,50-51
enable jumbo-frame ports all
create vlan "leaf3_trunk1"
create vlan "leaf3_trunk2"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 49 auto off speed 10000 duplex full
configure ports 50 auto off speed 10000 duplex full
configure ports 51 auto off speed 10000 duplex full
configure ports 52 auto off speed 10000 duplex full
configure ports 54 auto off speed 10000 duplex full
configure ports 55 auto off speed 10000 duplex full
configure ports 56 auto off speed 10000 duplex full
configure vlan Default add ports 2-49,52-56 untagged
configure vlan leaf3_trunk1 add ports 51 untagged
configure vlan leaf3_trunk2 add ports 50 untagged
configure vlan tenant add ports 1 tagged
configure vlan untagtenant add ports 1 untagged
configure vlan leaf3_trunk1 ipaddress 192.168.6.0 255.255.255.254
enable ipforwarding vlan leaf3_trunk1
configure vlan leaf3_trunk2 ipaddress 192.168.7.0 255.255.255.254
enable ipforwarding vlan leaf3_trunk2
configure vlan loop ipaddress 5.5.5.5 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.17 255.255.255.0
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan untagtenant ipaddress

# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"

#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 5.5.5.5 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
```

```
configure virtual-network "vni2" add vlan untagtenant

#
# Module bgp configuration.
#
configure bgp AS-number 10000
configure bgp routerid 5.5.5.5
configure bgp maximum-paths 2
configure bgp restart both
configure bgp add network 5.5.5.5/32
create bgp neighbor 3.3.3.100 remote-AS-number 10000
configure bgp neighbor 3.3.3.100 source-interface ipaddress 5.5.5.5
enable bgp neighbor 3.3.3.100
create bgp neighbor 4.4.4.4 remote-AS-number 10000
configure bgp neighbor 4.4.4.4 source-interface ipaddress 5.5.5.5
enable bgp neighbor 4.4.4.4
enable bgp neighbor 3.3.3.100 capability l2vpn-EVPN
enable bgp neighbor 4.4.4.4 capability l2vpn-EVPN
enable bgp
#
# Module ospf configuration.
#
enable ospf
configure ospf add vlan leaf1_trunk2 area 0.0.0.0 link-type point-to-point
configure ospf add vlan leaf3_trunk2 area 0.0.0.0 link-type point-to-point
configure ospf add vlan loop area 0.0.0.0 link-type point-to-point


#
# Module vrrp configuration.
#
create vrrp vlan tenant vrid 1
configure vrrp vlan tenant vrid 1 priority 255
create vrrp vlan untagtenant vrid 1
configure vrrp vlan untagtenant vrid 1 priority 255
configure vrrp vlan tenant vrid 1 add 10.1.100.100
configure vrrp vlan untagtenant vrid 1 add 20.1.100.100
enable vrrp vlan tenant vrid 1
enable vrrp vlan untagtenant vrid 1
```

Spine 1

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-128
configure vr VR-Default add ports 1-128
configure vlan default delete ports 33,57,89,105
enable jumbo-frame ports all
create vlan "leaf3_trunk1"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "routed"
configure vlan routed tag 4030
create vlan "trunk1_3"
create vlan "trunk2_3"
configure vlan Default add ports 1-32,34-56,58-88,90-104,106-128 untagged
configure vlan leaf3_trunk1 add ports 89 untagged
configure vlan routed add ports 33 tagged
configure vlan trunk1_3 add ports 57 untagged
configure vlan trunk2_3 add ports 105 untagged
configure vlan leaf3_trunk1 ipaddress 192.168.6.1 255.255.255.254
enable ipforwarding vlan leaf3_trunk1
```

```
configure vlan loop ipaddress 3.3.3.100 255.0.0.0
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.24 255.255.255.0
configure vlan routed ipaddress 172.16.1.1 255.255.255.0
enable ipforwarding vlan routed
configure vlan trunk1_3 ipaddress 192.168.1.1 255.255.255.254
enable ipforwarding vlan trunk1_3
configure vlan trunk2_3 ipaddress 192.168.3.1 255.255.255.254
enable ipforwarding vlan trunk2_3

#

#
# Module ospf configuration.
#
enable ospf
configure ospf add vlan leaf3_trunk1 area 0.0.0.0 link-type point-to-point
configure ospf add vlan loop area 0.0.0.0 link-type point-to-point
configure ospf add vlan routed area 0.0.0.0
configure ospf add vlan trunk1_3 area 0.0.0.0 link-type point-to-point
configure ospf add vlan trunk2_3 area 0.0.0.0 link-type point-to-point
#
# Module bgp configuration.
#
configure bgp AS-number 10000
configure bgp routerid 3.3.3.3
configure bgp maximum-paths 2
configure bgp add network 3.3.3.100/32
create bgp neighbor 1.1.1.1 remote-AS-number 10000
configure bgp neighbor 1.1.1.1 route-reflector-client
configure bgp neighbor 1.1.1.1 source-interface ipaddress 3.3.3.100
enable bgp neighbor 1.1.1.1
create bgp neighbor 2.2.2.2 remote-AS-number 10000
configure bgp neighbor 2.2.2.2 route-reflector-client
configure bgp neighbor 2.2.2.2 source-interface ipaddress 3.3.3.100
enable bgp neighbor 2.2.2.2
create bgp neighbor 5.5.5.5 remote-AS-number 10000
configure bgp neighbor 5.5.5.5 route-reflector-client
configure bgp neighbor 5.5.5.5 source-interface ipaddress 3.3.3.100
enable bgp neighbor 5.5.5.5
enable bgp neighbor 1.1.1.1 capability l2vpn-EVPN
enable bgp neighbor 2.2.2.2 capability l2vpn-EVPN
enable bgp neighbor 5.5.5.5 capability l2vpn-EVPN
enable bgp export ospf-extern1 address-family ipv4-unicast
enable bgp export ospf-extern2 address-family ipv4-unicast
enable bgp export ospf-inter address-family ipv4-unicast
enable bgp export ospf-intra address-family ipv4-unicast
enable bgp
```

Spine 2

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-64
configure vr VR-Default add ports 1-64
configure vlan default delete ports 25,27,49
create vlan "leaf3_trunk2"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "trunk1_4"
create vlan "trunk2_4"
configure ports 1 auto off speed 10000 duplex full
```

```
configure ports 2 auto off speed 10000 duplex full
configure ports 3 auto off speed 10000 duplex full
configure ports 4 auto off speed 10000 duplex full
configure ports 5 auto off speed 10000 duplex full
configure ports 6 auto off speed 10000 duplex full
configure ports 7 auto off speed 10000 duplex full
configure ports 8 auto off speed 10000 duplex full
configure ports 9 auto off speed 10000 duplex full
configure ports 10 auto off speed 10000 duplex full
configure ports 11 auto off speed 10000 duplex full
configure ports 12 auto off speed 10000 duplex full
configure ports 13 auto off speed 10000 duplex full
configure ports 14 auto off speed 10000 duplex full
configure ports 15 auto off speed 10000 duplex full
configure ports 16 auto off speed 10000 duplex full
configure ports 17 auto off speed 10000 duplex full
configure ports 18 auto off speed 10000 duplex full
configure ports 19 auto off speed 10000 duplex full
configure ports 20 auto off speed 10000 duplex full
configure ports 21 auto off speed 10000 duplex full
configure ports 22 auto off speed 10000 duplex full
configure ports 23 auto off speed 10000 duplex full
configure ports 24 auto off speed 10000 duplex full
configure ports 25 auto off speed 10000 duplex full
configure ports 26 auto off speed 10000 duplex full
configure ports 27 auto off speed 10000 duplex full
configure ports 28 auto off speed 10000 duplex full
configure ports 29 auto off speed 10000 duplex full
configure ports 30 auto off speed 10000 duplex full
configure ports 31 auto off speed 10000 duplex full
configure ports 32 auto off speed 10000 duplex full
configure ports 33 auto off speed 10000 duplex full
configure ports 34 auto off speed 10000 duplex full
configure ports 35 auto off speed 10000 duplex full
configure ports 36 auto off speed 10000 duplex full
configure ports 37 auto off speed 10000 duplex full
configure ports 38 auto off speed 10000 duplex full
configure ports 39 auto off speed 10000 duplex full
configure ports 40 auto off speed 10000 duplex full
configure ports 41 auto off speed 10000 duplex full
configure ports 42 auto off speed 10000 duplex full
configure ports 43 auto off speed 10000 duplex full
configure ports 44 auto off speed 10000 duplex full
configure ports 45 auto off speed 10000 duplex full
configure ports 46 auto off speed 10000 duplex full
configure ports 47 auto off speed 10000 duplex full
configure ports 48 auto off speed 10000 duplex full
configure vlan Default add ports 1-24,26,28-48,50-64 untagged
configure vlan leaf3_trunk2 add ports 25 untagged
configure vlan trunk1_4 add ports 27 untagged
configure vlan trunk2_4 add ports 49 untagged
configure vlan leaf3_trunk2 ipaddress 192.168.7.1 255.255.255.254
enable ipforwarding vlan leaf3_trunk2
configure vlan loop ipaddress 4.4.4.4 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.13.10 255.255.255.0
configure vlan trunk1_4 ipaddress 192.168.2.1 255.255.255.254
enable ipforwarding vlan trunk1_4
configure vlan trunk2_4 ipaddress 192.168.4.1 255.255.255.254
enable ipforwarding vlan trunk2_4

#
# Module bgp configuration.
#
```

```
configure bgp AS-number 10000
configure bgp routerid 4.4.4.4
configure bgp maximum-paths 2
configure bgp add network 3.3.3.100/32
configure bgp add network 4.4.4.4/32
create bgp neighbor 1.1.1.1 remote-AS-number 10000
configure bgp neighbor 1.1.1.1 route-reflector-client
configure bgp neighbor 1.1.1.1 source-interface ipaddress 4.4.4.4
enable bgp neighbor 1.1.1.1
create bgp neighbor 2.2.2.2 remote-AS-number 10000
configure bgp neighbor 2.2.2.2 route-reflector-client
configure bgp neighbor 2.2.2.2 source-interface ipaddress 4.4.4.4
enable bgp neighbor 2.2.2.2
create bgp neighbor 5.5.5.5 remote-AS-number 10000
configure bgp neighbor 5.5.5.5 route-reflector-client
configure bgp neighbor 5.5.5.5 source-interface ipaddress 4.4.4.4
enable bgp neighbor 5.5.5.5
enable bgp neighbor 1.1.1.1 capability l2vpn-EVPN
enable bgp neighbor 2.2.2.2 capability l2vpn-EVPN
enable bgp neighbor 5.5.5.5 capability l2vpn-EVPN
enable bgp
#
# Module ospf configuration.
#
enable ospf
configure ospf add vlan leaf3_trunk2 area 0.0.0.0 link-type point-to-point
configure ospf add vlan loop area 0.0.0.0
configure ospf add vlan trunk1_4 area 0.0.0.0 link-type point-to-point
configure ospf add vlan trunk2_4 area 0.0.0.0 link-type point-to-point
```

Verification

```
Leaf1.1 # show ospf neighbor
Neighbor ID     Pri State              Up/Dead Time             Address         Interface
          BFD Session State
=============================================================================================
2.2.2.2           1 FULL      /DROTHER 00:00:59:02/00:00:00:08  192.168.5.1     trunk1_2
          None

192.168.6.1       1 FULL      /DROTHER 00:00:59:02/00:00:00:02  192.168.1.1     trunk1_3
          None

192.168.7.1       1 FULL      /DROTHER 00:00:59:02/00:00:00:00  192.168.2.1     trunk1_4
          None

Total number of neighbors: 3 (All neighbors in Full state)

Leaf2.15 # show ospf neighbor
Neighbor ID     Pri State              Up/Dead Time             Address         Interface
          BFD Session State
=============================================================================================
1.1.1.1           1 FULL      /DROTHER 00:00:59:22/00:00:00:02  192.168.5.0     trunk1_2
          None

192.168.6.1       1 FULL      /DROTHER 00:00:59:37/00:00:00:05  192.168.3.1     trunk2_3
          None

192.168.7.1       1 FULL      /DROTHER 00:00:59:37/00:00:00:06  192.168.4.1     trunk2_4
          None

Leaf3.12 # show ospf neighbor
Neighbor ID     Pri State              Up/Dead Time             Address         Interface
          BFD Session State
=============================================================================================
```

```
192.168.6.1       1 FULL      /DROTHER 00:00:02:00/00:00:00:01  192.168.6.1     leaf3_tru
          None

192.168.7.1       1 FULL      /DROTHER 00:01:33:25/00:00:00:01  192.168.7.1     leaf3_tru
          None
Total number of neighbors: 2 (All neighbors in Full state)

Spine1.1 # show ospf neighbor
Neighbor ID     Pri State              Up/Dead Time             Address         Interface
          BFD Session State
================================================================================================
192.168.7.0       1 FULL      /DROTHER 00:00:22:40/00:00:00:01  192.168.6.0     leaf3_tru
          None

1.1.1.1           1 FULL      /DROTHER 00:01:23:37/00:00:00:07  192.168.1.0     trunk1_3
          None

2.2.2.2           1 FULL      /DROTHER 00:01:23:53/00:00:00:03  192.168.3.0     trunk2_3
          None

Total number of neighbors: 3 (All neighbors in Full state)

Spine2.1 # show ospf neighbor
Neighbor ID     Pri State              Up/Dead Time             Address         Interface
          BFD Session State
================================================================================================
192.168.7.0       1 FULL      /DROTHER 00:01:54:24/00:00:00:04  192.168.7.0     leaf3_tru
          None

1.1.1.1           1 FULL      /DROTHER 00:01:23:55/00:00:00:05  192.168.2.0     trunk1_4
          None

2.2.2.2           1 FULL      /DROTHER 00:01:24:10/00:00:00:10  192.168.4.0     trunk2_4
          None

Total number of neighbors: 3 (All neighbors in Full state)
```

Verify BGP Adjacency

```
Leaf1.1 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------------------------
------------------
Ie-- 2.2.2.2                                 10000      1      ESTABLISHED  150    149
(0    ) 0:1:59:37
Ie-- 3.3.3.100                               10000      1      ESTABLISHED  182    149
(0    ) 0:1:59:37
Ie-- 4.4.4.4                                 10000      1      ESTABLISHED  191    158
(0    ) 0:0:17:47

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 0                IBGP Peers          : 3
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0

(Software Update Required) Leaf2.16 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
```

```
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------------
------------------
Ie-- 1.1.1.1                                10000      1      ESTABLISHED  271    288
(0    ) 0:1:59:59
Ie-- 3.3.3.100                              10000      1      ESTABLISHED  556    415
(0    ) 0:2:52:09
Ie-- 4.4.4.4                                10000      1      ESTABLISHED  1585   1509
(0    ) 0:20:58:35

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 0                IBGP Peers          : 3
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0

Leaf3.1 # show bgp neighbor

     Peer                                   AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------------
------------------
Ie-- 3.3.3.100                              10000      1      ESTABLISHED  171    144
(0    ) 0:1:55:46
Ie-- 4.4.4.4                                10000      1      ESTABLISHED  175    144
(0    ) 0:1:55:37

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 2
  EBGP Peers        : 0                IBGP Peers          : 2
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 2                Disabled            : 0

Spine1.2 # show bgp neighbor

     Peer                                   AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------------
------------------
Ier- 1.1.1.1                                10000      1      ESTABLISHED  150    183
(0    ) 0:2:00:42
Ier- 2.2.2.2                                10000      1      ESTABLISHED  222    265
(0    ) 0:2:52:53
Ier- 5.5.5.5                                10000      1      ESTABLISHED  184    249
(0    ) 0:1:56:07

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 0                IBGP Peers          : 3
  RR Client         : 3                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0

Spine2.2 # show bgp neighbor

     Peer                                   AS         Weight State        InMsgs
```

```
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------------
------------------
Ier- 1.1.1.1                                  10000      1      ESTABLISHED  276    390
(0    ) 0:0:19:14
Ier- 2.2.2.2                                  10000      1      ESTABLISHED  1501   1606
(0    ) 0:20:59:40
Ier- 5.5.5.5                                  10000      1      ESTABLISHED  1465   1595
(0    ) 0:1:56:20

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 0                IBGP Peers          : 3
  RR Client         : 3                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0
```

Verify EVPN EVI

```
Leaf1.2 # show bgp EVPN evi
Flgs EVI-Idx Name                         VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*         10                             100 1.1.1.200               1.1.1.200:10
*
    10000:10             10000:10
*
10000:268435556      10000:268435556
*
10000:1073741834     10000:1073741834
*       4087                           200 1.1.1.200             1.1.1.200:4087
*
  10000:4087           10000:4087
*
10000:268435656      10000:268435656
*
10000:1073745911     10000:1073745911

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 0   Dyn: 2

Leaf2.17 # show bgp EVPN evi
Flgs EVI-Idx Name                         VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*         10                             100 1.1.1.200               1.1.1.200:10
*
    10000:10             10000:10
*
10000:268435556      10000:268435556
*
10000:1073741834     10000:1073741834
*       4088                           200 1.1.1.200             1.1.1.200:4088
```

```
*
  10000:4088           10000:4088
*
10000:268435656      10000:268435656
*
10000:1073745912     10000:1073745912

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 0   Dyn: 2

Leaf3.2 # show bgp EVPN evi
Flgs EVI-Idx Name                         VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*         10                            100 5.5.5.5                   5.5.5.5:10
*
    10000:10             10000:10
*
10000:268435556      10000:268435556
*
10000:1073741834     10000:1073741834
*       4090                          200 5.5.5.5                 5.5.5.5:4090
*
  10000:4090           10000:4090
*
10000:268435656      10000:268435656
*
10000:1073745914     10000:1073745914

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 0   Dyn: 2

Spine1.3 # show bgp EVPN evi
Flgs EVI-Idx Name                         VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*         10                            100 3.3.3.100               3.3.3.100:10
*
    10000:10             10000:10
*
10000:268435556      10000:268435556
*
10000:1073741834     10000:1073741834
*       4091                          200 3.3.3.100             3.3.3.100:4091
*
  10000:4091           10000:4091
*
10000:268435656      10000:268435656
```

```
*
10000:1073745915     10000:1073745915

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 0   Dyn: 2
```

Verify IMR

```
Leaf1.4 # show bgp routes l2vpn-EVPN inclusive-multicast all

Routes:
     Originating IP      Tunnel ID           VNI
Peer                                 Next-Hop            AS-Path
-------------------------------------------------------------------------------------------
-----------------------------------
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
2.2.2.2                              1.1.1.200
  ?  RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
2.2.2.2                              1.1.1.200
  ?  RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
3.3.3.100                            1.1.1.200
  ?  RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
4.4.4.4                              1.1.1.200
  ?  RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6
     3.3.3.100           3.3.3.100           100
3.3.3.100                            3.3.3.100
  ?  RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6
     3.3.3.100           3.3.3.100           200
3.3.3.100                            3.3.3.100
  ?  RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             100
3.3.3.100                            5.5.5.5
  ?  RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             100
4.4.4.4                              5.5.5.5
  ?  RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             200
3.3.3.100                            5.5.5.5
  ?  RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             200
4.4.4.4                              5.5.5.5

Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
       (s) Stale, (m) Multipath, (u) Unfeasible

Origin: (?) Incomplete, (e) EGP, (i) IGP

BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 10
  Feasible Routes   : 10
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
```

```
Route Statistics on Session Type
  Routes from Int Peer: 10
  Routes from Ext Peer: 0

Leaf2.18 #  show bgp routes l2vpn-EVPN inclusive-multicast all

Routes:
     Originating IP      Tunnel ID           VNI
Peer                                 Next-Hop            AS-Path
-------------------------------------------------------------------------------------------
-----------------------------------
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
1.1.1.1                              1.1.1.200
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
3.3.3.100                            1.1.1.200
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
4.4.4.4                              1.1.1.200
  ?  RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
1.1.1.1                              1.1.1.200
  ?  RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
3.3.3.100                            1.1.1.200
  ?  RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
4.4.4.4                              1.1.1.200
  ?  RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6
     3.3.3.100           3.3.3.100           100
3.3.3.100                            3.3.3.100
  ?  RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6
     3.3.3.100           3.3.3.100           200
3.3.3.100                            3.3.3.100
  ?  RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             100
3.3.3.100                            5.5.5.5
  ?  RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             100
4.4.4.4                              5.5.5.5
  ?  RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             200
3.3.3.100                            5.5.5.5
  ?  RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             200
4.4.4.4                              5.5.5.5

Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
       (s) Stale, (m) Multipath, (u) Unfeasible

Origin: (?) Incomplete, (e) EGP, (i) IGP

BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 12
  Feasible Routes   : 12
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
Route Statistics on Session Type
  Routes from Int Peer: 12
  Routes from Ext Peer: 0

Leaf3.12 # show bgp routes l2vpn-EVPN inclusive-multicast all
```

```
Routes:
     Originating IP      Tunnel ID           VNI
Peer                                 Next-Hop            AS-Path
-------------------------------------------------------------------------------------------
-----------------------------------
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
3.3.3.100                            1.1.1.200
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
4.4.4.4                              1.1.1.200
  ?  RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
3.3.3.100                            1.1.1.200
  ?  RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
4.4.4.4                              1.1.1.200
  ?  RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
3.3.3.100                            1.1.1.200
  ?  RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
4.4.4.4                              1.1.1.200
  ?  RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6
     3.3.3.100           3.3.3.100           100
3.3.3.100                            3.3.3.100
  ?  RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6
     3.3.3.100           3.3.3.100           200
3.3.3.100                            3.3.3.100

Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
       (s) Stale, (m) Multipath, (u) Unfeasible

Origin: (?) Incomplete, (e) EGP, (i) IGP

BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 8
  Feasible Routes   : 8
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
Route Statistics on Session Type
  Routes from Int Peer: 8
  Routes from Ext Peer: 0

show bgp routes l2vpn-EVPN inclusive-multicast all

Routes:
     Originating IP      Tunnel ID           VNI
Peer                                 Next-Hop            AS-Path
-------------------------------------------------------------------------------------------
-----------------------------------
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
1.1.1.1                              1.1.1.200
  ?  RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           100
2.2.2.2                              1.1.1.200
  ?  RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
1.1.1.1                              1.1.1.200
  ?  RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6
     1.1.1.200           1.1.1.200           200
```

```
2.2.2.2                                1.1.1.200
  ?  RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             100
5.5.5.5                                5.5.5.5
  ?  RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6
     5.5.5.5             5.5.5.5             200
5.5.5.5                                5.5.5.5

Flags: (*) Preferred BGP route, (>) Active, (d) Suppressed, (h) History
       (s) Stale, (m) Multipath, (u) Unfeasible

Origin: (?) Incomplete, (e) EGP, (i) IGP

BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 6
  Feasible Routes   : 6
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
Route Statistics on Session Type
  Routes from Int Peer: 6
  Routes from Ext Peer: 0
```

Verify Type 2 MAC Routes

```
Leaf1.4 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
L        10 00:00:0b:cb:21:a4
100                                          Yes
L        10 00:00:0c:14:88:4b
100                                          Yes
L        10 00:00:0c:34:c1:ed
100                                          Yes
L        10 00:04:96:9d:42:32
100                                          Yes
L        10 00:04:96:a3:fb:b0
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4087 00:00:00:1c:7e:5f
200                                          Yes
R      4087 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4087 00:00:04:d6:4d:6e
200                                          Yes
L      4087 00:00:04:d6:4d:6f
200                                          Yes
L      4087 00:04:96:9d:42:32
200                                          Yes
L      4087 00:04:96:a3:fb:b0
200                                          Yes
R      4087 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 14
```

```
Leaf2.19 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
L        10 00:00:0b:cb:21:a4
100                                          Yes
L        10 00:00:0c:14:88:4b
100                                          Yes
L        10 00:00:0c:34:c1:ed
100                                          Yes
L        10 00:04:96:9d:42:32
100                                          Yes
L        10 00:04:96:a3:fb:b0
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4088 00:00:00:1c:7e:5f
200                                          Yes
R      4088 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4088 00:00:04:d6:4d:6e
200                                          Yes
L      4088 00:00:04:d6:4d:6f
200                                          Yes
L      4088 00:04:96:9d:42:32
200                                          Yes
L      4088 00:04:96:a3:fb:b0
200                                          Yes
R      4088 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 14

Leaf3.21 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
L        10 00:11:88:ff:04:74
100                                          Yes
R      4090 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
```

```
R      4090 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4090 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
L      4090 00:11:88:ff:04:74
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 19

Spine1.37 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
L        10 00:00:00:1c:7e:60
100                                          Yes
R        10 00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
R      4091 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4091 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
L      4091 00:00:00:1c:7e:61
200                                          Yes
R      4091 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 00:04:96:a3:fb:b0 1.1.1.200
```

```
200                                          Yes
R      4091 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 19
```

Verify Type 2 IPv4 Routes

```
Leaf1.6 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
L        10 10.1.100.1      00:00:0b:cb:21:a4
100                                          Yes
L        10 10.1.100.2      00:00:0c:34:c1:ed
100                                          Yes
L        10 10.1.100.100    00:04:96:a3:fb:b0
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
R      4087 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4087 20.1.100.1      00:00:04:d6:4d:6e
200                                          Yes
L      4087 20.1.100.2      00:00:04:d6:4d:6f
200                                          Yes
L      4087 20.1.100.100    00:04:96:a3:fb:b0
200                                          Yes
R      4087 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 10

Leaf2.21 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
L        10 10.1.100.5      00:00:0c:14:88:4b
100                                          Yes
L        10 10.1.100.2      00:00:0c:34:c1:ed
100                                          Yes
L        10 10.1.100.100    00:04:96:9d:42:32
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4088 20.1.100.5      00:00:00:1c:7e:5f
200                                          Yes
R      4088 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4088 20.1.100.2      00:00:04:d6:4d:6f
200                                          Yes
L      4088 20.1.100.100    00:04:96:9d:42:32
```

```
200                                          Yes
R      4088 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 10

Leaf3.22 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 10.1.100.1      00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 10.1.100.5      00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 10.1.100.2      00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
L        10 10.1.100.100    00:11:88:ff:04:74
100                                          Yes
R      4090 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4090 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 20.1.100.100    00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 20.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
L      4090 20.1.100.100    00:11:88:ff:04:74
200                                          Yes

Spine1.38 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
L        10 10.1.100.3      00:00:00:1c:7e:60
100                                          Yes
R        10 10.1.100.1      00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 10.1.100.5      00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 10.1.100.2      00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
```

```
R      4091 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
L      4091 20.1.100.3      00:00:00:1c:7e:61
200                                          Yes
R      4091 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 20.1.100.100    00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 20.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.
```

Verify MAC and IPARP Entries

```
Leaf1.8 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    7      NO  tenant        10    16
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    7      NO  tenant        10    21
VR-Default    10.1.100.3       00:00:00:1c:7e:60    0     YES  tenant        10    VR-
Default:3.3.3.100
VR-Default    10.1.100.5       00:00:0c:14:88:4b    0      NO  tenant        10    1
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    192.168.1.1      00:04:96:9c:2d:24   20      NO  trunk1_3      4089  5
VR-Default    192.168.2.1      00:04:96:9b:b7:c9    8      NO  trunk1_4      4088  7
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    7      NO  untagtenant   4087  16
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    7      NO  untagtenant   4087  21
VR-Default    20.1.100.3       00:00:00:1c:7e:61    0     YES  untagtenant   4087  VR-
Default:3.3.3.100
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    0      NO  untagtenant   4087  1
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4087  VR-
Default:5.5.5.5
VR-Default    192.168.5.1      00:04:96:9d:42:32   20      NO  trunk1_2      4001  21
VR-Default    172.16.1.1       00:04:96:9d:42:32   20      NO  isc           4000  21

Dynamic Entries  :          10             Static Entries            :          4
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

(pacman debug) Leaf1.8 # show fdb
MAC                                   VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                    untagtenant(4087) 0000  d mi       S   1
```

```
00:00:00:1c:7e:60                          tenant(0010) 0000  s m        SXE VR-
Default:3.3.3.100
00:00:00:1c:7e:61                       untagtenant(4087) 0000  s m        SXE VR-
Default:3.3.3.100
00:00:00:47:ce:72                          tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:00:47:ce:73                       untagtenant(4087) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:04:d6:4d:6e                       untagtenant(4087) 0000  d mi           16
00:00:04:d6:4d:6f                       untagtenant(4087) 0000  d mi       S   21
00:00:0b:cb:21:a4                          tenant(0010) 0000  d mi           16
00:00:0c:14:88:4b                          tenant(0010) 0000  d mi       S   1
00:00:0c:34:c1:ed                          tenant(0010) 0000  d mi       S   21
00:04:96:9b:b7:c9                        trunk1_4(4088) 0000  d mi           7
00:04:96:9c:2d:24                        trunk1_3(4089) 0000  d mi           5
00:04:96:9d:42:32                          tenant(0010) 0000  s m        S   21
00:04:96:9d:42:32                       untagtenant(4087) 0000  s m        S   21
00:04:96:9d:42:32                        trunk1_2(4001) 0000  s m        S   21
00:04:96:9d:42:32                             isc(4000) 0000  s m        S   21
00:04:96:af:9d:1c                         Default(0001) 0008  d m            33
00:11:88:ff:04:74                          tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:11:88:ff:04:74                       untagtenant(4087) 0000  s m        SXE VR-
Default:5.5.5.5

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 19 Static: 10  Perm: 0  Dyn: 9  Dropped: 0  Locked: 0  Locked with Timeout: 0

(Software Update Required) Leaf2.21 # show fdb
MAC                                     VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                       untagtenant(4088) 0000  d mi       S   53
00:00:00:1c:7e:60                          tenant(0010) 0000  s m        SXE VR-
Default:3.3.3.100
00:00:00:1c:7e:61                       untagtenant(4088) 0000  s m        SXE VR-
Default:3.3.3.100
00:00:00:47:ce:72                          tenant(0010) 0000  s mi       SXE VR-
Default:5.5.5.5
00:00:00:47:ce:73                       untagtenant(4088) 0000  s mi       SXE VR-
Default:5.5.5.5
00:00:04:d6:4d:6e                       untagtenant(4088) 0000  d mi       S   66
00:00:04:d6:4d:6f                       untagtenant(4088) 0000  d mi           4
00:00:0b:cb:21:a4                          tenant(0010) 0000  d mi       S   66
00:00:0c:14:88:4b                          tenant(0010) 0000  d mi       S   53
00:00:0c:34:c1:ed                          tenant(0010) 0000  d mi           4
00:04:96:9b:b7:c9                        trunk2_4(4089) 0000  d mi           49
00:04:96:9c:2d:24                        trunk2_3(4090) 0000  d mi           57
00:04:96:a3:fb:b0                          tenant(0010) 0000  s m        S   66
00:04:96:a3:fb:b0                       untagtenant(4088) 0000  s m        S   66
00:04:96:a3:fb:b0                        trunk1_2(4001) 0000  s m        S   66
00:04:96:a3:fb:b0                             isc(4000) 0000  s m        S   66
00:11:88:ff:04:74                          tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:11:88:ff:04:74                       untagtenant(4088) 0000  s m        SXE VR-
Default:5.5.5.5
```

```
Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 18 Static: 10  Perm: 0  Dyn: 8  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
(Software Update Required) Leaf2.22 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    0      NO  tenant        10    66
VR-Default    10.1.100.2       00:00:0c:34:c1:ed   31      NO  tenant        10    4
VR-Default    10.1.100.3       00:00:00:1c:7e:60    0     YES  tenant        10    VR-
Default:3.3.3.100
VR-Default    10.1.100.5       00:00:0c:14:88:4b    7      NO  tenant        10    53
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    192.168.3.1      00:04:96:9c:2d:24   60      NO  trunk2_3      4090  57
VR-Default    192.168.4.1      00:04:96:9b:b7:c9   11      NO  trunk2_4      4089  49
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    0      NO  untagtenant   4088  66
VR-Default    20.1.100.2       00:00:04:d6:4d:6f   37      NO  untagtenant   4088  4
VR-Default    20.1.100.3       00:00:00:1c:7e:61    0     YES  untagtenant   4088  VR-
Default:3.3.3.100
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    7      NO  untagtenant   4088  53
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4088  VR-
Default:5.5.5.5
VR-Default    192.168.5.0      00:04:96:a3:fb:b0   60      NO  trunk1_2      4001  66
VR-Default    172.16.1.0       00:04:96:a3:fb:b0   60      NO  isc           4000  66

Dynamic Entries  :          10             Static Entries            :          4
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          60 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :     2700000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

(Software Update Required) Leaf3.23 # show fdb
MAC                                   VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                    untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:00:00:1c:7e:60                         tenant(0010) 0000  s m         XE VR-
Default:3.3.3.100
00:00:00:1c:7e:61                    untagtenant(4090) 0000  s m         XE VR-
Default:3.3.3.100
00:00:00:47:ce:72                         tenant(0010) 0000  d mi           1
00:00:00:47:ce:73                    untagtenant(4090) 0000  d mi           1
00:00:04:d6:4d:6e                    untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:00:04:d6:4d:6f                    untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
```

```
00:00:0b:cb:21:a4                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0c:14:88:4b                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0c:34:c1:ed                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:9b:b7:c9                    leaf3_trunk2(4093) 0000  d mi           50
00:04:96:9c:2d:24                    leaf3_trunk1(4094) 0000  d mi           51
00:04:96:9d:42:32                          tenant(0010) 0051  d m         X  VR-
Default:3.3.3.100
00:04:96:9d:42:32                     untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                     untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 16 Static: 11  Perm: 0  Dyn: 5  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
(Software Update Required) Leaf3.24 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    192.168.6.1      00:04:96:9c:2d:24   20      NO  leaf3_trunk1  4094  51
VR-Default    192.168.7.1      00:04:96:9b:b7:c9   20      NO  leaf3_trunk2  4093  50
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.3       00:00:00:1c:7e:60    0     YES  tenant        10    VR-
Default:3.3.3.100
VR-Default    10.1.100.5       00:00:0c:14:88:4b    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.6       00:00:00:47:ce:72    0      NO  tenant        10    1
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.3       00:00:00:1c:7e:61    0     YES  untagtenant   4090  VR-
Default:3.3.3.100
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.6       00:00:00:47:ce:73    0      NO  untagtenant   4090  1

Dynamic Entries  :           4             Static Entries           :          8
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh              :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning  :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256
```

```
# show fdb
MAC                                    VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                     untagtenant(4091) 0000  s m         XE VR-
Default:1.1.1.200
00:00:00:1c:7e:60                          tenant(0010) 0006  d m            33
00:00:00:1c:7e:61                     untagtenant(4091) 0006  d m            33
00:00:00:47:ce:72                          tenant(0010) 0000  s m         XE VR-
Default:5.5.5.5
00:00:00:47:ce:73                     untagtenant(4091) 0000  s m         XE VR-
Default:5.5.5.5
00:00:01:23:c7:5f                          routed(4030) 0000  d mi           33
00:00:04:d6:4d:6e                     untagtenant(4091) 0000  s m         XE VR-
Default:1.1.1.200
00:00:04:d6:4d:6f                     untagtenant(4091) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0b:cb:21:a4                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0c:14:88:4b                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0c:34:c1:ed                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:5e:00:01:01                          tenant(0010) 0000  d m         X  VR-
Default:1.1.1.200
00:00:5e:00:01:01                     untagtenant(4091) 0000  d m         X  VR-
Default:1.1.1.200
00:04:96:9d:42:32                        trunk2_3(4093) 0000  d mi           105
00:04:96:9d:42:32                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:9d:42:32                     untagtenant(4091) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                        trunk1_3(4094) 0000  d mi           57
00:04:96:a3:fb:b0                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                     untagtenant(4091) 0000  s m         XE VR-
Default:1.1.1.200
00:11:88:ff:04:74                          tenant(0010) 0000  s m         XE VR-
Default:5.5.5.5
00:11:88:ff:04:74                     untagtenant(4091) 0000  s m         XE VR-
Default:5.5.5.5
00:11:88:ff:04:74                    leaf3_trunk1(4088) 0000  d mi           89

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 22 Static: 14  Perm: 0  Dyn: 8  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
(pacman debug) Spine1.45 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    192.168.1.0      00:04:96:a3:fb:b0   20      NO  trunk1_3      4094  57
VR-Default    192.168.3.0      00:04:96:9d:42:32   20      NO  trunk2_3      4093  105
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.3       00:00:00:1c:7e:60    1      NO  tenant        10    33
```

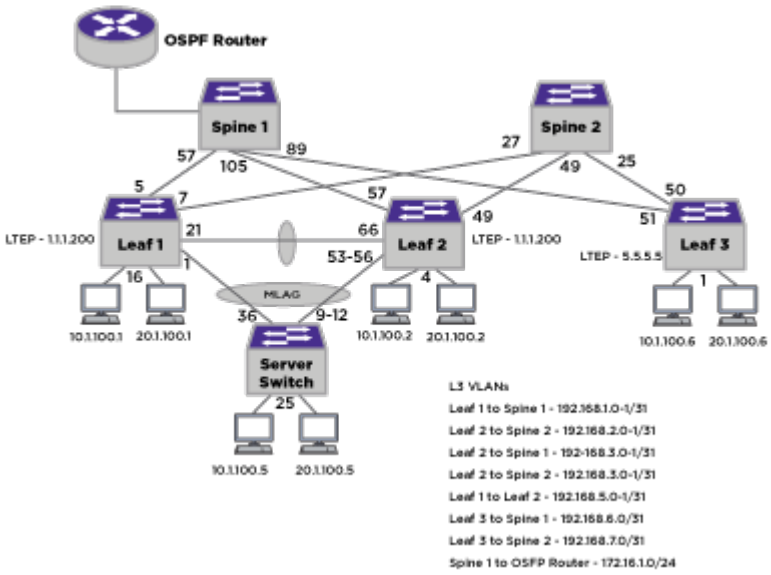
```
VR-Default    10.1.100.5       00:00:0c:14:88:4b    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    10.1.100.100     00:11:88:ff:04:74    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    0     YES  untagtenant   4091  VR-
Default:1.1.1.200
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    0     YES  untagtenant   4091  VR-
Default:1.1.1.200
VR-Default    20.1.100.3       00:00:00:1c:7e:61    1      NO  untagtenant   4091  33
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    0     YES  untagtenant   4091  VR-
Default:1.1.1.200
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4091  VR-
Default:5.5.5.5
VR-Default    20.1.100.100     00:11:88:ff:04:74    0     YES  untagtenant   4091  VR-
Default:5.5.5.5
VR-Default    172.16.1.2       00:00:01:23:c7:5f   15      NO  routed        4030  33
VR-Default    192.168.6.0      00:11:88:ff:04:74   20      NO  leaf3_trunk1  4088  89

Dynamic Entries  :           6             Static Entries            :         10
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256
```

Apply IP Anycast

Configure and apply IP Anycast on Leaf1, Leaf2, and Leaf3:

```
# configure ip anycast mac 02:e0:52:00:01:00
# configure vlan tenant ipaddress 224.0.0.18/32
# enable ipforwarding broadcast vlan tenant
# enable ip anycast vlan tenant
```

EVPN with eBGP Configuration Example



Configuration Notes

- Configure and apply IP Anycast on leafs (see IP and MAC Anycast on page 2228, and configure IP address on tenant VLAN and enable IP forwarding.
- EVI with manual RT is configured. Here is an example:

```
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
```

- MLAG peers share the same LTEP IP, but use different BGP router IDs.
- BGP adjacency between MLAG peers is recommended for alternate path (additional L3 VLAN between MLAG peers).
- OSPF routes are exported to BGP on spine1.
- Single-hop eBGP is used for underlay and overlay.
- On spines, the “Next-Hop-Unchnaged” option is used for l2vpn-EVPN address family.
- BGP fast-external-fallover is enabled on all nodes.
- Static LACP MAC is configured on MLAG peers.
- BGP maximum paths is set to 2 for this topology and IP route sharing is enabled.
- ARP suppression is enabled on tenant VLANs.

Note
On SLX and other platforms “retain route-targets” should be enabled on spine/transit L3 nodes. ExtremeXOSby default retains all RTs and does not have a command to tun it off.

Note

Using auto-peering negates the need for manual BGP and EVI configuration. BGP Auto-peering can be created with the command `create auto-peering bgp vlans 4010-4020 routerid 1.1.1.100 AS-number 10000`:

- 4010-4020 VLANs are automatically created by BGP Auto-peering. Ports are added to these VLANs as untagged.
- BGP Auto-peering uses eBGP—all nodes in auto-peering domain need to use different AS numbers.
- On the MLAG peers—ISC ports should be tagged on all user-created VLANs.

Leaf 1

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-36
configure vr VR-Default add ports 1-36
configure vlan default delete ports 1,4-5,7,16,19,21
enable jumbo-frame ports all
create vlan "isc"
configure vlan isc tag 4000
create vlan "loop"
enable loopback-mode vlan loop
create vlan "mlagvtep"
enable loopback-mode vlan mlagvtep
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "trunk1_2"
configure vlan trunk1_2 tag 4001
create vlan "trunk1_3"
create vlan "trunk1_4"
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 1 auto off speed 10000 duplex full
configure ports 2 auto off speed 10000 duplex full
configure ports 3 auto off speed 10000 duplex full
configure ports 4 auto off speed 10000 duplex full
configure ports 5 auto off speed 10000 duplex full
configure ports 6 auto off speed 10000 duplex full
configure ports 7 auto off speed 10000 duplex full
configure ports 8 auto off speed 10000 duplex full
configure ports 9 auto off speed 10000 duplex full
configure ports 10 auto off speed 10000 duplex full
configure ports 11 auto off speed 10000 duplex full
configure ports 12 auto off speed 10000 duplex full
configure ports 13 auto off speed 10000 duplex full
configure ports 14 auto off speed 10000 duplex full
configure ports 15 auto off speed 10000 duplex full
configure ports 16 auto off speed 10000 duplex full
configure ports 17 auto off speed 10000 duplex full
configure ports 18 auto off speed 10000 duplex full
configure ports 19 auto off speed 10000 duplex full
configure ports 20 auto off speed 10000 duplex full
configure ports 21 auto off speed 10000 duplex full
configure ports 22 auto off speed 10000 duplex full
configure ports 23 auto off speed 10000 duplex full
configure ports 24 auto off speed 10000 duplex full
```

```
configure ports 26 auto off speed 10000 duplex full
configure ports 27 auto off speed 10000 duplex full
configure ports 28 auto off speed 10000 duplex full
enable sharing 1 grouping 1 algorithm address-based L2 lacp
configure vlan Default add ports 2-3,6,8-15,17-18,20,22-36 untagged
configure vlan isc add ports 21 tagged
configure vlan tenant add ports 1,16,21 tagged
configure vlan trunk1_2 add ports 21 tagged
configure vlan trunk1_3 add ports 5 untagged
configure vlan trunk1_4 add ports 7 untagged
configure vlan untagtenant add ports 1,16,21 untagged
configure vlan isc ipaddress 172.16.1.0 255.255.255.254
configure vlan loop ipaddress 1.1.1.1 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.27 255.255.255.0
configure vlan mlagvtep ipaddress 1.1.1.200 255.255.255.255
enable ipforwarding vlan mlagvtep
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan trunk1_2 ipaddress 192.168.5.0 255.255.255.254
enable ipforwarding vlan trunk1_2
configure vlan trunk1_3 ipaddress 192.168.1.0 255.255.255.254
enable ipforwarding vlan trunk1_3
configure vlan trunk1_4 ipaddress 192.168.2.0 255.255.255.254
enable ipforwarding vlan trunk1_4
configure vlan untagtenant ipaddress 20.1.100.100 255.255.255.0
enable ipforwarding vlan untagtenant

#
# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"

#
# Module vsm configuration.
#
create mlag peer "leaf2"
configure mlag peer "leaf2" ipaddress 172.16.1.1 vr VR-Default
configure mlag peer "leaf2" lacp-mac 00:00:de:ad:be:ef
enable mlag port 1 peer "leaf2" id 1

#
# Module bgp configuration.
#
configure bgp AS-number 10000
configure bgp routerid 1.1.1.1
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 1.1.1.1/32
configure bgp add network 1.1.1.200/32
create bgp neighbor 192.168.1.1 remote-AS-number 30000
enable bgp neighbor 192.168.1.1
create bgp neighbor 192.168.2.1 remote-AS-number 40000
enable bgp neighbor 192.168.2.1
create bgp neighbor 192.168.5.1 remote-AS-number 20000
enable bgp neighbor 192.168.5.1
enable bgp neighbor 192.168.5.1 address-family l2vpn-evpn next-hop-unchanged
enable bgp neighbor 192.168.1.1 capability l2vpn-EVPN
enable bgp neighbor 192.168.2.1 capability l2vpn-EVPN
enable bgp neighbor 192.168.5.1 capability l2vpn-EVPN
enable bgp export direct address-family ipv4-unicast
enable bgp
create bgp EVPN instance evi_1
```

```
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.1.1.200 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant
```

Leaf 2

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-72
configure vr VR-Default add ports 1-72
configure vlan default delete ports 1-72
enable jumbo-frame ports all
create vlan "isc"
configure vlan isc tag 4000
create vlan "loop"
enable loopback-mode vlan loop
create vlan "loop2"
enable loopback-mode vlan loop2
create vlan "mlagvtep"
enable loopback-mode vlan mlagvtep
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "trunk1_2"
configure vlan trunk1_2 tag 4001
create vlan "trunk2_3"
create vlan "trunk2_4"
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 50 auto off speed 10000 duplex full
configure ports 51 auto off speed 10000 duplex full
configure ports 52 auto off speed 10000 duplex full
enable sharing 53 grouping 53-56 algorithm address-based L2 lacp
configure vlan isc add ports 66 tagged
configure vlan tenant add ports 4-5,53,66 tagged
configure vlan trunk1_2 add ports 66 tagged
configure vlan trunk2_3 add ports 57 untagged
configure vlan trunk2_4 add ports 49 untagged
configure vlan untagtenant add ports 4-5,53,66 untagged
configure vlan isc ipaddress 172.16.1.1 255.255.255.254
configure vlan loop ipaddress 2.2.2.2 255.255.255.255
enable ipforwarding vlan loop
configure vlan loop2 ipaddress 2.2.2.100 255.255.255.255
enable ipforwarding vlan loop2
configure vlan Mgmt ipaddress 10.127.16.19 255.255.255.0
configure vlan mlagvtep ipaddress 1.1.1.200 255.0.0.0
enable ipforwarding vlan mlagvtep
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
```

```
configure vlan trunk1_2 ipaddress 192.168.5.1 255.255.255.254
enable ipforwarding vlan trunk1_2
configure vlan trunk2_3 ipaddress 192.168.3.0 255.255.255.254
enable ipforwarding vlan trunk2_3
configure vlan trunk2_4 ipaddress 192.168.4.0 255.255.255.254
enable ipforwarding vlan trunk2_4
configure vlan untagtenant ipaddress 20.1.100.100 255.255.255.0
enable ipforwarding vlan untagtenant
#
# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"


#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.1.1.200 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant

#
# Module bgp configuration.
#
configure bgp AS-number 20000
configure bgp routerid 2.2.2.2
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 1.1.1.200/32
configure bgp add network 2.2.2.2/32
create bgp neighbor 192.168.3.1 remote-AS-number 30000
enable bgp neighbor 192.168.3.1
create bgp neighbor 192.168.4.1 remote-AS-number 40000
enable bgp neighbor 192.168.4.1
create bgp neighbor 192.168.5.0 remote-AS-number 10000
enable bgp neighbor 192.168.5.0
enable bgp neighbor 192.168.5.0 address-family l2vpn-evpn next-hop-unchanged
enable bgp neighbor 192.168.3.1 capability l2vpn-EVPN
enable bgp neighbor 192.168.4.1 capability l2vpn-EVPN
enable bgp neighbor 192.168.5.0 capability l2vpn-EVPN
enable bgp export direct address-family ipv4-unicast
enable bgp
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
#
# Module vrrp configuration.
#
create vrrp vlan tenant vrid 1
configure vrrp vlan tenant vrid 1 priority 255
create vrrp vlan untagtenant vrid 1
configure vrrp vlan untagtenant vrid 1 priority 255
configure vrrp vlan tenant vrid 1 add 10.1.100.100
configure vrrp vlan untagtenant vrid 1 add 20.1.100.100
enable vrrp vlan tenant vrid 1
enable vrrp vlan untagtenant vrid 1
```

```
#
# Module vsm configuration.
#
create mlag peer "leaf1"
configure mlag peer "leaf1" ipaddress 172.16.1.0 vr VR-Default
configure mlag peer "leaf1" lacp-mac 00:00:de:ad:be:ef
enable mlag port 53 peer "leaf1" id 1
```

Leaf 3

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-56
configure vr VR-Default add ports 1-56
configure vlan default delete ports 1,50-51
enable jumbo-frame ports all
create vlan "leaf3_trunk1"
create vlan "leaf3_trunk2"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 49 auto off speed 10000 duplex full
configure ports 50 auto off speed 10000 duplex full
configure ports 51 auto off speed 10000 duplex full
configure ports 52 auto off speed 10000 duplex full
configure ports 54 auto off speed 10000 duplex full
configure ports 55 auto off speed 10000 duplex full
configure ports 56 auto off speed 10000 duplex full
configure vlan Default add ports 2-49,52-56 untagged
configure vlan leaf3_trunk1 add ports 51 untagged
configure vlan leaf3_trunk2 add ports 50 untagged
configure vlan tenant add ports 1 tagged
configure vlan untagtenant add ports 1 untagged
configure vlan leaf3_trunk1 ipaddress 192.168.6.0 255.255.255.254
enable ipforwarding vlan leaf3_trunk1
configure vlan leaf3_trunk2 ipaddress 192.168.7.0 255.255.255.254
enable ipforwarding vlan leaf3_trunk2
configure vlan loop ipaddress 5.5.5.5 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.17 255.255.255.0
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan untagtenant ipaddress

# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"

#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 5.5.5.5 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
```

```
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant

#
# Module bgp configuration.
#
configure bgp AS-number 50000
configure bgp routerid 5.5.5.5
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp restart both
configure bgp add network 5.5.5.5/32
create bgp neighbor 192.168.6.1 remote-AS-number 30000
enable bgp neighbor 192.168.6.1
create bgp neighbor 192.168.7.1 remote-AS-number 40000
enable bgp neighbor 192.168.7.1
enable bgp neighbor 192.168.6.1 capability l2vpn-EVPN
enable bgp neighbor 192.168.7.1 capability l2vpn-EVPN
enable bgp
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200

#
# Module vrrp configuration.
#
create vrrp vlan tenant vrid 1
configure vrrp vlan tenant vrid 1 priority 255
create vrrp vlan untagtenant vrid 1
configure vrrp vlan untagtenant vrid 1 priority 255
configure vrrp vlan tenant vrid 1 add 10.1.100.100
configure vrrp vlan untagtenant vrid 1 add 20.1.100.100
enable vrrp vlan tenant vrid 1
enable vrrp vlan untagtenant vrid 1
```

Spine 1

```
configure vlan default delete ports all
configure vr VR-Default delete ports 1-128
configure vr VR-Default add ports 1-128
configure vlan default delete ports 33,57,89,105
enable jumbo-frame ports all
create vlan "leaf3_trunk1"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "routed"
configure vlan routed tag 4030
create vlan "trunk1_3"
create vlan "trunk2_3"
configure vlan Default add ports 1-32,34-56,58-88,90-104,106-128 untagged
configure vlan leaf3_trunk1 add ports 89 untagged
configure vlan routed add ports 33 tagged
configure vlan trunk1_3 add ports 57 untagged
configure vlan trunk2_3 add ports 105 untagged
configure vlan leaf3_trunk1 ipaddress 192.168.6.1 255.255.255.254
enable ipforwarding vlan leaf3_trunk1
configure vlan loop ipaddress 3.3.3.100 255.0.0.0
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.24 255.255.255.0
configure vlan routed ipaddress 172.16.1.1 255.255.255.0
enable ipforwarding vlan routed
```

```
configure vlan trunk1_3 ipaddress 192.168.1.1 255.255.255.254
enable ipforwarding vlan trunk1_3
configure vlan trunk2_3 ipaddress 192.168.3.1 255.255.255.254
enable ipforwarding vlan trunk2_3

#
# Module bgp configuration.
#
configure bgp AS-number 30000
configure bgp routerid 3.3.3.3
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 3.3.3.100/32
create bgp neighbor 192.168.1.0 remote-AS-number 10000
enable bgp neighbor 192.168.1.0
create bgp neighbor 192.168.3.0 remote-AS-number 20000
enable bgp neighbor 192.168.3.0
create bgp neighbor 192.168.6.0 remote-AS-number 50000
enable bgp neighbor 192.168.6.0
enable bgp neighbor 192.168.1.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.1.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.3.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.3.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.6.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.6.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp export ospf-extern1 address-family ipv4-unicast
enable bgp export ospf-extern2 address-family ipv4-unicast
enable bgp export ospf-inter address-family ipv4-unicast
enable bgp export ospf-intra address-family ipv4-unicast
enable bgp
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
#
# Module ospf configuration.
#
enable ospf
configure ospf add vlan routed area 0.0.0.0
```

Spine 2

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-64
configure vr VR-Default add ports 1-64
configure vlan default delete ports 25,27,49
create vlan "leaf3_trunk2"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "trunk1_4"
create vlan "trunk2_4"
configure ports 1 auto off speed 10000 duplex full
configure ports 2 auto off speed 10000 duplex full
configure ports 3 auto off speed 10000 duplex full
configure ports 4 auto off speed 10000 duplex full
configure ports 5 auto off speed 10000 duplex full
configure ports 6 auto off speed 10000 duplex full
configure ports 7 auto off speed 10000 duplex full
configure ports 8 auto off speed 10000 duplex full
```

```
configure ports 9 auto off speed 10000 duplex full
configure ports 10 auto off speed 10000 duplex full
configure ports 11 auto off speed 10000 duplex full
configure ports 12 auto off speed 10000 duplex full
configure ports 13 auto off speed 10000 duplex full
configure ports 14 auto off speed 10000 duplex full
configure ports 15 auto off speed 10000 duplex full
configure ports 16 auto off speed 10000 duplex full
configure ports 17 auto off speed 10000 duplex full
configure ports 18 auto off speed 10000 duplex full
configure ports 19 auto off speed 10000 duplex full
configure ports 20 auto off speed 10000 duplex full
configure ports 21 auto off speed 10000 duplex full
configure ports 22 auto off speed 10000 duplex full
configure ports 23 auto off speed 10000 duplex full
configure ports 24 auto off speed 10000 duplex full
configure ports 25 auto off speed 10000 duplex full
configure ports 26 auto off speed 10000 duplex full
configure ports 27 auto off speed 10000 duplex full
configure ports 28 auto off speed 10000 duplex full
configure ports 29 auto off speed 10000 duplex full
configure ports 30 auto off speed 10000 duplex full
configure ports 31 auto off speed 10000 duplex full
configure ports 32 auto off speed 10000 duplex full
configure ports 33 auto off speed 10000 duplex full
configure ports 34 auto off speed 10000 duplex full
configure ports 35 auto off speed 10000 duplex full
configure ports 36 auto off speed 10000 duplex full
configure ports 37 auto off speed 10000 duplex full
configure ports 38 auto off speed 10000 duplex full
configure ports 39 auto off speed 10000 duplex full
configure ports 40 auto off speed 10000 duplex full
configure ports 41 auto off speed 10000 duplex full
configure ports 42 auto off speed 10000 duplex full
configure ports 43 auto off speed 10000 duplex full
configure ports 44 auto off speed 10000 duplex full
configure ports 45 auto off speed 10000 duplex full
configure ports 46 auto off speed 10000 duplex full
configure ports 47 auto off speed 10000 duplex full
configure ports 48 auto off speed 10000 duplex full
configure vlan Default add ports 1-24,26,28-48,50-64 untagged
configure vlan leaf3_trunk2 add ports 25 untagged
configure vlan trunk1_4 add ports 27 untagged
configure vlan trunk2_4 add ports 49 untagged
configure vlan leaf3_trunk2 ipaddress 192.168.7.1 255.255.255.254
enable ipforwarding vlan leaf3_trunk2
configure vlan loop ipaddress 4.4.4.4 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.13.10 255.255.255.0
configure vlan trunk1_4 ipaddress 192.168.2.1 255.255.255.254
enable ipforwarding vlan trunk1_4
configure vlan trunk2_4 ipaddress 192.168.4.1 255.255.255.254
enable ipforwarding vlan trunk2_4
#
# Module bgp configuration.
#
configure bgp AS-number 40000
configure bgp routerid 4.4.4.4
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 3.3.3.100/32
configure bgp add network 4.4.4.4/32
create bgp neighbor 192.168.2.0 remote-AS-number 10000
enable bgp neighbor 192.168.2.0
```

```
create bgp neighbor 192.168.4.0 remote-AS-number 20000
enable bgp neighbor 192.168.4.0
create bgp neighbor 192.168.7.0 remote-AS-number 50000
enable bgp neighbor 192.168.7.0
enable bgp neighbor 192.168.2.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.2.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.4.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.4.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.7.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.7.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp
```

Verification

Verify BGP Adjacency

```
Leaf1.2 #  show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.1.1                             30000      1      ESTABLISHED  72     72
(0    ) 0:0:14:38
Ee-- 192.168.2.1                             40000      1      ESTABLISHED  48     78
(0    ) 0:0:14:53
Ee-- 192.168.5.1                             20000      1      ESTABLISHED  60     61
(0    ) 0:0:14:44

Leaf2.2 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.3.1                             30000      1      ESTABLISHED  56     79
(0    ) 0:0:14:44
Ee-- 192.168.4.1                             40000      1      ESTABLISHED  70     76
(0    ) 0:0:14:41
Ee-- 192.168.5.0                             10000      1      ESTABLISHED  61     60
(0    ) 0:0:14:50

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 3                IBGP Peers          : 0
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0
(Software Update Required) Leaf3.9 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.6.1                             30000      1      ESTABLISHED  74     81
(0    ) 0:0:14:47
Ee-- 192.168.7.1                             40000      1      ESTABLISHED  76     79
(0    ) 0:0:14:56
Spine1.3 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
```

```
-------------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.1.0                              10000      1      ESTABLISHED  74     75
(0    ) 0:0:17:03
Ee-- 192.168.3.0                              20000      1      ESTABLISHED  81     59
(0    ) 0:0:17:03
Ee-- 192.168.6.0                              50000      1      ESTABLISHED  83     77
(0    ) 0:0:17:03

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 3                IBGP Peers          : 0
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0

Spine2.4 # show bgp neighbor

     Peer                                     AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.2.0                              10000      1      ESTABLISHED  245    212
(0    ) 0:0:17:34
Ee-- 192.168.4.0                              20000      1      ESTABLISHED  246    232
(0    ) 0:0:17:15
Ee-- 192.168.7.0                              50000      1      ESTABLISHED  273    277
(0    ) 0:0:17:27

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 3                IBGP Peers          : 0
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0
```

Verify EVI

```
show bgp EVPN evi
Flgs EVI-Idx Name                         VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*S        10                 evi_1        100 1.1.1.200               1.1.1.200:10
*S
     100:100              100:100
*
    10000:10             10000:10
*
10000:268435556      10000:268435556
*
10000:1073741834     10000:1073741834
*S      4087                 evi_2        200 1.1.1.200             1.1.1.200:4087
*S
     200:200              200:200
*
  10000:4087           10000:4087
*
10000:268435656      10000:268435656
*
```

```
10000:1073745911     10000:1073745911

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 2   Dyn: 0
Leaf2.3 # show bgp EVPN evi
Flgs EVI-Idx Name                          VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*S        10                 evi_1        100 1.1.1.200               1.1.1.200:10
*S
     100:100              100:100
*
    20000:10             20000:10
*
20000:268435556      20000:268435556
*
20000:1073741834     20000:1073741834
*S      4088                 evi_2        200 1.1.1.200             1.1.1.200:4088
*S
     200:200              200:200
*
  20000:4088           20000:4088
*
20000:268435656      20000:268435656
*
20000:1073745912     20000:1073745912

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 2   Dyn: 0

Leaf3.10 # show bgp EVPN evi
Flgs EVI-Idx Name                          VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*S        10                 evi_1        100 5.5.5.5                   5.5.5.5:10
*S
     100:100              100:100
*
    50000:10             50000:10
*
50000:268435556      50000:268435556
*
50000:1073741834     50000:1073741834
*S      4090                 evi_2        200 5.5.5.5                 5.5.5.5:4090
*S
     200:200              200:200
*
  50000:4090           50000:4090
```

```
*
50000:268435656      50000:268435656
*
50000:1073745914     50000:1073745914

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 2   Dyn: 0

Spine1.4 # show bgp EVPN evi
Flgs EVI-Idx Name                         VNI Local VTEP
RD            Import RT            Export RT
---- ------- -------------------- ---------- --------------- --------------------
-------------------- --------------------
*S        10                 evi_1        100 3.3.3.100               3.3.3.100:10
*S
     100:100              100:100
*
    30000:10             30000:10
*
30000:268435556      30000:268435556
*
30000:1073741834     30000:1073741834
*S      4091                 evi_2        200 3.3.3.100             3.3.3.100:4091
*S
     200:200              200:200
*
  30000:4091           30000:4091
*
30000:268435656      30000:268435656
*
30000:1073745915     30000:1073745915

Flags: (*) Entry is active, (S) Entry is statically configured.
RD: Route Distinguisher
RT: Route Target

Some static entries may have multiple Route Target sets. These are listed in rows
immediately following the initial row
for the EVI-Idx.

EVIs Displayed - Total: 2   Static: 2   Dyn: 0
```

Verify IMR

```
Leaf1.4 # show bgp routes l2vpn-EVPN inclusive-multicast detail all

Routes:
RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.5.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000
Ext Community: rt:100:100 rt:20000:10 rt:20000:1073741834 encap:8 rt:20000:268435556


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.1.1
```

```
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 30000 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.2.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 40000 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.5.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656


RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.1.1
Origin Incomplete, Next-Hop 3.3.3.100, LPref 100, MED 0
Weight 0,
As-PATH: 30000
Ext Community: rt:100:100 rt:30000:10 rt:30000:1073741834 encap:8 rt:30000:268435556


RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.5.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000 30000
Ext Community: rt:100:100 rt:30000:10 rt:30000:1073741834 encap:8 rt:30000:268435556


RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.1.1
Origin Incomplete, Next-Hop 3.3.3.100, LPref 100, MED 0
Weight 0,
As-PATH: 30000
Ext Community: rt:200:200 rt:30000:4091 rt:30000:1073745915 encap:8 rt:30000:268435656


RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.5.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000 30000
Ext Community: rt:200:200 rt:30000:4091 rt:30000:1073745915 encap:8 rt:30000:268435656


RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.1.1
Origin Incomplete, Next-Hop 5.5.5.5, LPref 100, MED 0
Weight 0,
As-PATH: 30000 50000
Ext Community: rt:100:100 rt:50000:10 rt:50000:1073741834 encap:8 rt:50000:268435556


RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.2.1
```

```
Origin Incomplete, Next-Hop 5.5.5.5, LPref 100, MED 0
Weight 0,
As-PATH: 40000 50000
Ext Community: rt:100:100 rt:50000:10 rt:50000:1073741834 encap:8 rt:50000:268435556


RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.5.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000 30000 50000
Ext Community: rt:100:100 rt:50000:10 rt:50000:1073741834 encap:8 rt:50000:268435556


RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.1.1
Origin Incomplete, Next-Hop 5.5.5.5, LPref 100, MED 0
Weight 0,
As-PATH: 30000 50000
Ext Community: rt:200:200 rt:50000:4090 rt:50000:1073745914 encap:8 rt:50000:268435656


RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.2.1
Origin Incomplete, Next-Hop 5.5.5.5, LPref 100, MED 0
Weight 0,
As-PATH: 40000 50000
Ext Community: rt:200:200 rt:50000:4090 rt:50000:1073745914 encap:8 rt:50000:268435656


RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.5.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000 30000 50000
Ext Community: rt:200:200 rt:50000:4090 rt:50000:1073745914 encap:8 rt:50000:268435656



BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 14
  Feasible Routes   : 14
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
Route Statistics on Session Type
  Routes from Int Peer: 0
  Routes from Ext Peer: 14
Leaf3.12 # show bgp routes l2vpn-EVPN inclusive-multicast detail all

Routes:
RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.6.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 30000 10000
Ext Community: rt:100:100 rt:10000:10 rt:10000:1073741834 encap:8 rt:10000:268435556


RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.7.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 40000 10000
```

```
Ext Community: rt:100:100 rt:10000:10 rt:10000:1073741834 encap:8 rt:10000:268435556


RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.6.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 30000 10000
Ext Community: rt:200:200 rt:10000:4087 rt:10000:1073745911 encap:8 rt:10000:268435656


RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.7.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 40000 10000
Ext Community: rt:200:200 rt:10000:4087 rt:10000:1073745911 encap:8 rt:10000:268435656


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.6.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 30000 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.7.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 40000 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656


RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.6.1
Origin Incomplete, Next-Hop 3.3.3.100, LPref 100, MED 0
Weight 0,
As-PATH: 30000
Ext Community: rt:100:100 rt:30000:10 rt:30000:1073741834 encap:8 rt:30000:268435556


RD: 3.3.3.100:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.7.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 40000 10000 30000
Ext Community: rt:100:100 rt:30000:10 rt:30000:1073741834 encap:8 rt:30000:268435556


RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.6.1
Origin Incomplete, Next-Hop 3.3.3.100, LPref 100, MED 0
Weight 0,
As-PATH: 30000
Ext Community: rt:200:200 rt:30000:4091 rt:30000:1073745915 encap:8 rt:30000:268435656


RD: 3.3.3.100:4091     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 3.3.3.100           Tunnel ID: 3.3.3.100            Peer 192.168.7.1
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 40000 10000 30000
```

```
Ext Community: rt:200:200 rt:30000:4091 rt:30000:1073745915 encap:8 rt:30000:268435656



BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 10
  Feasible Routes   : 10
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
Route Statistics on Session Type
  Routes from Int Peer: 0
  Routes from Ext Peer: 10

Spine1.5 # show bgp routes l2vpn-EVPN inclusive-multicast detail all

Routes:
RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.1.0
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 10000
Ext Community: rt:100:100 rt:10000:10 rt:10000:1073741834 encap:8 rt:10000:268435556


RD: 1.1.1.200:10       Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.3.0
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000
Ext Community: rt:100:100 rt:20000:10 rt:20000:1073741834 encap:8 rt:20000:268435556


RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.1.0
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 10000
Ext Community: rt:200:200 rt:10000:4087 rt:10000:1073745911 encap:8 rt:10000:268435656


RD: 1.1.1.200:4087     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.3.0
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000 10000
Ext Community: rt:200:200 rt:10000:4087 rt:10000:1073745911 encap:8 rt:10000:268435656


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.1.0
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 10000 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656


RD: 1.1.1.200:4088     Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 1.1.1.200           Tunnel ID: 1.1.1.200            Peer 192.168.3.0
Origin Incomplete, Next-Hop 1.1.1.200, LPref 100, MED 0
Weight 0,
As-PATH: 20000
Ext Community: rt:200:200 rt:20000:4088 rt:20000:1073745912 encap:8 rt:20000:268435656
```

```
RD: 5.5.5.5:10         Ethernet Tag: 0     Tunnel Type: 6     VNI: 100
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.6.0
Origin Incomplete, Next-Hop 5.5.5.5, LPref 100, MED 0
Weight 0,
As-PATH: 50000
Ext Community: rt:100:100 rt:50000:10 rt:50000:1073741834 encap:8 rt:50000:268435556


RD: 5.5.5.5:4090       Ethernet Tag: 0     Tunnel Type: 6     VNI: 200
Originating IP: 5.5.5.5             Tunnel ID: 5.5.5.5              Peer 192.168.6.0
Origin Incomplete, Next-Hop 5.5.5.5, LPref 100, MED 0
Weight 0,
As-PATH: 50000
Ext Community: rt:200:200 rt:50000:4090 rt:50000:1073745914 encap:8 rt:50000:268435656

BGP Route Statistics EVPN route type 3
  Total Rxed Routes : 8
  Feasible Routes   : 8
  Active Routes     : 0
  Rejected Routes   : 0
  Unfeasible Routes : 0
Route Statistics on Session Type
  Routes from Int Peer: 0
  Routes from Ext Peer: 8
```

Verify type 2 MAC Routes

```
Leaf1.5 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 00:00:0b:cb:21:a4
100                                          Yes
L        10 00:00:0c:14:88:4b
100                                          Yes
L        10 00:00:0c:34:c1:ed
100                                          Yes
L        10 00:04:96:9d:42:32
100                                          Yes
L        10 00:04:96:a3:fb:b0
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4087 00:00:00:1c:7e:5f
200                                          Yes
R      4087 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4087 00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4087 00:00:04:d6:4d:6e
200                                          Yes
L      4087 00:00:04:d6:4d:6f
200                                          Yes
L      4087 00:04:96:9d:42:32
200                                          Yes
L      4087 00:04:96:a3:fb:b0
200                                          Yes
R      4087 00:11:88:ff:04:74 5.5.5.5
```

```
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 16

Leaf2.5 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 00:00:0b:cb:21:a4
100                                          Yes
L        10 00:00:0c:14:88:4b
100                                          Yes
L        10 00:00:0c:34:c1:ed
100                                          Yes
L        10 00:04:96:9d:42:32
100                                          Yes
L        10 00:04:96:a3:fb:b0
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4088 00:00:00:1c:7e:5f
200                                          Yes
R      4088 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4088 00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4088 00:00:04:d6:4d:6e
200                                          Yes
L      4088 00:00:04:d6:4d:6f
200                                          Yes
L      4088 00:04:96:9d:42:32
200                                          Yes
L      4088 00:04:96:a3:fb:b0
200                                          Yes
R      4088 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 16
Leaf3.12 #  show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
L        10 00:00:00:47:ce:72
100                                          Yes
R        10 00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
```

```
R        10 00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
L        10 00:11:88:ff:04:74
100                                          Yes
R      4090 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4090 00:00:00:47:ce:73
200                                          Yes
R      4090 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4090 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
L      4090 00:11:88:ff:04:74
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 21

Spine1.7 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
L        10 00:00:00:1c:7e:60
100                                          Yes
R        10 00:00:00:47:ce:72 5.5.5.5
100                                          Yes
R        10 00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
R      4091 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4091 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
L      4091 00:00:00:1c:7e:61
```

```
200                                          Yes
R      4091 00:00:00:47:ce:73 5.5.5.5
200                                          Yes
R      4091 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 21
```

Verify Type 2 IPv4 Routes

```
Leaf1.6 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
----------------------------- ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 10.1.100.6      00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 10.1.100.1      00:00:0b:cb:21:a4
100                                          Yes
L        10 10.1.100.5      00:00:0c:14:88:4b
100                                          Yes
L        10 10.1.100.100    00:04:96:a3:fb:b0
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4087 20.1.100.5      00:00:00:1c:7e:5f
200                                          Yes
R      4087 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4087 20.1.100.6      00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4087 20.1.100.1      00:00:04:d6:4d:6e
200                                          Yes
L      4087 20.1.100.2      00:00:04:d6:4d:6f
200                                          Yes
L      4087 20.1.100.100    00:04:96:a3:fb:b0
200                                          Yes
R      4087 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.
```

```
Total MAC/IP entries displayed: 13

Leaf2.6 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
R        10 10.1.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                           Yes
R        10 10.1.1.100.6      00:00:00:47:ce:72 5.5.5.5
100                                           Yes
L        10 10.1.1.100.1      00:00:0b:cb:21:a4
100                                           Yes
L        10 10.1.1.100.5      00:00:0c:14:88:4b
100                                           Yes
L        10 10.1.1.100.2      00:00:0c:34:c1:ed
100                                           Yes
L        10 10.1.1.100.100    00:04:96:9d:42:32
100                                           Yes
R        10 10.1.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                           Yes
L      4088 20.1.1.100.5      00:00:00:1c:7e:5f
200                                           Yes
R      4088 20.1.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                           Yes
R      4088 20.1.1.100.6      00:00:00:47:ce:73 5.5.5.5
200                                           Yes
L      4088 20.1.1.100.1      00:00:04:d6:4d:6e
200                                           Yes
L      4088 20.1.1.100.2      00:00:04:d6:4d:6f
200                                           Yes
L      4088 20.1.1.100.100    00:04:96:9d:42:32
200                                           Yes
R      4088 20.1.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                           Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 14

Leaf3.13 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
R        10 10.1.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                           Yes
L        10 10.1.1.100.6      00:00:00:47:ce:72
100                                           Yes
R        10 10.1.1.100.1      00:00:0b:cb:21:a4 1.1.1.200
100                                           Yes
R        10 10.1.1.100.5      00:00:0c:14:88:4b 1.1.1.200
100                                           Yes
R        10 10.1.1.100.2      00:00:0c:34:c1:ed 1.1.1.200
100                                           Yes
R        10 10.1.1.100.100    00:04:96:9d:42:32 1.1.1.200
100                                           Yes
R        10 10.1.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
100                                           Yes
L        10 10.1.1.100.100    00:11:88:ff:04:74
100                                           Yes
R      4090 20.1.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                           Yes
```

```
R      4090 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4090 20.1.100.6      00:00:00:47:ce:73
200                                          Yes
R      4090 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 20.1.100.100    00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 20.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
L      4090 20.1.100.100    00:11:88:ff:04:74
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 19

Spine1.13 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
L        10 10.1.100.3      00:00:00:1c:7e:60
100                                          Yes
R        10 10.1.100.6      00:00:00:47:ce:72 5.5.5.5
100                                          Yes
R        10 10.1.100.1      00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 10.1.100.5      00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 10.1.100.2      00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
R      4091 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4091 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
L      4091 20.1.100.3      00:00:00:1c:7e:61
200                                          Yes
R      4091 20.1.100.6      00:00:00:47:ce:73 5.5.5.5
200                                          Yes
R      4091 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 20.1.100.100    00:04:96:9d:42:32 1.1.1.200
```

```
200                                          Yes
R      4091 20.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 19
```

Verify ARP and FDB Entries

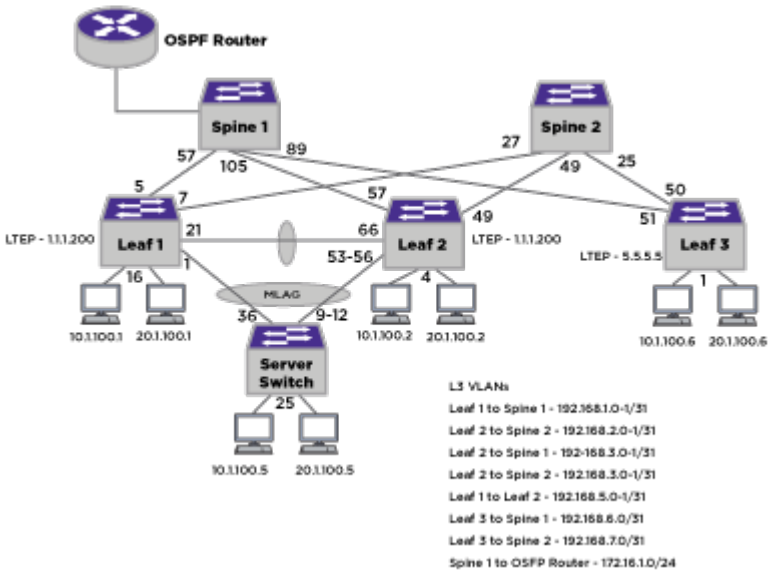
```
Leaf1.7 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    10.1.100.1       00:00:0b:cb:21:a4   17      NO  tenant        10    16
VR-Default    10.1.100.5       00:00:0c:14:88:4b    7      NO  tenant        10    1
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    192.168.1.1      00:04:96:9c:2d:24   20      NO  trunk1_3      4089  5
VR-Default    192.168.2.1      00:04:96:9b:b7:c9   20      NO  trunk1_4      4088  7
VR-Default    20.1.100.1       00:00:04:d6:4d:6e   12      NO  untagtenant   4087  16
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    7      NO  untagtenant   4087  21
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    7      NO  untagtenant   4087  1
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4087  VR-
Default:5.5.5.5
VR-Default    192.168.5.1      00:04:96:9d:42:32   20      NO  trunk1_2      4001  21
VR-Default    172.16.1.1       00:04:96:9d:42:32   20      NO  isc           4000  21

Dynamic Entries  :           9             Static Entries            :          2
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

Leaf1.8 #  show fdb
MAC                                    VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                    untagtenant(4087) 0000  d mi       S   1
00:00:00:47:ce:72                         tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:00:47:ce:73                    untagtenant(4087) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:04:d6:4d:6e                    untagtenant(4087) 0000  d mi           16
00:00:04:d6:4d:6f                    untagtenant(4087) 0000  d mi       S   21
00:00:0b:cb:21:a4                         tenant(0010) 0000  d mi           16
00:00:0c:14:88:4b                         tenant(0010) 0000  d mi       S   1
00:00:0c:34:c1:ed                         tenant(0010) 0078  d m        S   21
00:04:96:9b:b7:c9                       trunk1_4(4088) 0000  d mi           7
00:04:96:9c:2d:24                       trunk1_3(4089) 0000  d mi           5
00:04:96:9d:42:32                         tenant(0010) 0000  s m        S   21
00:04:96:9d:42:32                    untagtenant(4087) 0000  s m        S   21
00:04:96:9d:42:32                       trunk1_2(4001) 0000  s m        S   21
```

```
00:04:96:9d:42:32                              isc(4000) 0000  s m        S   21
00:11:88:ff:04:74                            tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:11:88:ff:04:74                       untagtenant(4087) 0000  s m        SXE VR-
Default:5.5.5.5

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 16 Static: 8  Perm: 0  Dyn: 8  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300

Leaf2.7 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    7      NO  tenant        10    66
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    7      NO  tenant        10    4
VR-Default    10.1.100.5       00:00:0c:14:88:4b   31      NO  tenant        10    53
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    192.168.3.1      00:04:96:9c:2d:24   33      NO  trunk2_3      4090  57
VR-Default    192.168.4.1      00:04:96:9b:b7:c9   33      NO  trunk2_4      4089  49
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    7      NO  untagtenant   4088  66
VR-Default    20.1.100.2       00:00:04:d6:4d:6f   31      NO  untagtenant   4088  4
VR-Default    20.1.100.5       00:00:00:1c:7e:5f   31      NO  untagtenant   4088  53
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4088  VR-
Default:5.5.5.5
VR-Default    192.168.5.0      00:04:96:a3:fb:b0   33      NO  trunk1_2      4001  66
VR-Default    172.16.1.0       00:04:96:a3:fb:b0   33      NO  isc           4000  66

Dynamic Entries  :          10             Static Entries            :          2
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          60 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :     2700000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

(Software Update Required) Leaf2.8 # show fdb
MAC                                    VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                   untagtenant(4088) 0000  d mi       S   53
00:00:00:47:ce:72                        tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:00:47:ce:73                   untagtenant(4088) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:04:d6:4d:6e                   untagtenant(4088) 0000  d mi       S   66
00:00:04:d6:4d:6f                   untagtenant(4088) 0000  d mi           4
00:00:0b:cb:21:a4                        tenant(0010) 0000  d mi       S   66
00:00:0c:14:88:4b                        tenant(0010) 0000  d mi       S   53
00:00:0c:34:c1:ed                        tenant(0010) 0000  d mi           4
```

```
00:04:96:9b:b7:c9                          trunk2_4(4089) 0000  d mi           49
00:04:96:9c:2d:24                          trunk2_3(4090) 0000  d mi           57
00:04:96:a3:fb:b0                            tenant(0010) 0000  s m        S   66
00:04:96:a3:fb:b0                       untagtenant(4088) 0000  s m        S   66
00:04:96:a3:fb:b0                          trunk1_2(4001) 0000  s m        S   66
00:04:96:a3:fb:b0                               isc(4000) 0000  s m        S   66
00:11:88:ff:04:74                            tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:11:88:ff:04:74                       untagtenant(4088) 0000  s m        SXE VR-
Default:5.5.5.5

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 16 Static: 8  Perm: 0  Dyn: 8  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300

Leaf3.14 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    192.168.6.1      00:04:96:9c:2d:24   20      NO  leaf3_trunk1  4094  51
VR-Default    192.168.7.1      00:04:96:9b:b7:c9   20      NO  leaf3_trunk2  4093  50
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.5       00:00:0c:14:88:4b    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.6       00:00:00:47:ce:72   15      NO  tenant        10    1
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.6       00:00:00:47:ce:73   20      NO  untagtenant   4090  1

Dynamic Entries  :           4             Static Entries            :          6
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

(Software Update Required) Leaf3.15 # show fdb
MAC                                     VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                       untagtenant(4090) 0000  s mi        XE VR-
Default:1.1.1.200
00:00:00:47:ce:72                            tenant(0010) 0000  d mi           1
```

```
00:00:00:47:ce:73                      untagtenant(4090) 0000  d mi           1
00:00:04:d6:4d:6e                      untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:00:04:d6:4d:6f                      untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0b:cb:21:a4                           tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0c:14:88:4b                           tenant(0010) 0000  s mi        XE VR-
Default:1.1.1.200
00:00:0c:34:c1:ed                           tenant(0010) 0000  s mi        XE VR-
Default:1.1.1.200
00:04:96:9b:b7:c9                    leaf3_trunk2(4093) 0000  d mi           50
00:04:96:9c:2d:24                    leaf3_trunk1(4094) 0000  d mi           51
00:04:96:9d:42:32                           tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:9d:42:32                      untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                           tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                      untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 14 Static: 10  Perm: 0  Dyn: 4  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
```

EVPN with eBGP Between Leaf and Spine Configuration Example



EVPN with eBGP between leaf and spine. Spines are on a dedicated ASn, Each leaf is on separate ASn. MLAG peers share ASn.

Configuration Notes

- Configure and apply IP Anycast on leafs (see IP and MAC Anycast on page 2228, and configure IP address on tenant VLAN and enable IP forwarding.
- EVI with manual RT is configured. Here is an example:

```
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
```

- MLAG peers share the same LTEP IP, ASNUM, but use different BGP router IDs.
- BGP adjacency between MLAG peers is recommended for alternate path. (additional L3 VLAN between MLAG peers).
- OSPF routes are exported to BGP on Spine1.
- eBGP between leaf and spines. MLAG peers use ASNUM 10000; spines use ASNUM 20000; Leaf 3 uses ASNUM 30000.
- Allow-as-in on leaf nodes (VTEPs). This is to ensure EVPN routes from other leaf nodes are not dropped due to BGP split horizon (ASPATH present in route).
- BGP fast-external-fallover is enabled on all nodes.
- Spines are configured with “next-hop-unchanged” for L2VPN-EVPN routes.
- Static LACP MAC is configured on MLAG peers.
- BGP maximum paths is set to 2 for this topology and IP route sharing is enabled.
- ARP suppression is enabled on tenant VLANs.

Note
On SLX and other platforms “retain route-targets” should be enabled on spine/transit L3 nodes. ExtremeXOS by default retains all RTs and does not have a command to turn it off.

Leaf 1

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-36
configure vr VR-Default add ports 1-36
configure vlan default delete ports 1,4-5,7,16,19,21
enable jumbo-frame ports all
create vlan "isc"
configure vlan isc tag 4000
create vlan "loop"
enable loopback-mode vlan loop
create vlan "mlagvtep"
enable loopback-mode vlan mlagvtep
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "trunk1_2"
configure vlan trunk1_2 tag 4001
create vlan "trunk1_3"
create vlan "trunk1_4"
```

```
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 1 auto off speed 10000 duplex full
configure ports 2 auto off speed 10000 duplex full
configure ports 3 auto off speed 10000 duplex full
configure ports 4 auto off speed 10000 duplex full
configure ports 5 auto off speed 10000 duplex full
configure ports 6 auto off speed 10000 duplex full
configure ports 7 auto off speed 10000 duplex full
configure ports 8 auto off speed 10000 duplex full
configure ports 9 auto off speed 10000 duplex full
configure ports 10 auto off speed 10000 duplex full
configure ports 11 auto off speed 10000 duplex full
configure ports 12 auto off speed 10000 duplex full
configure ports 13 auto off speed 10000 duplex full
configure ports 14 auto off speed 10000 duplex full
configure ports 15 auto off speed 10000 duplex full
configure ports 16 auto off speed 10000 duplex full
configure ports 17 auto off speed 10000 duplex full
configure ports 18 auto off speed 10000 duplex full
configure ports 19 auto off speed 10000 duplex full
configure ports 20 auto off speed 10000 duplex full
configure ports 21 auto off speed 10000 duplex full
configure ports 22 auto off speed 10000 duplex full
configure ports 23 auto off speed 10000 duplex full
configure ports 24 auto off speed 10000 duplex full
configure ports 26 auto off speed 10000 duplex full
configure ports 27 auto off speed 10000 duplex full
configure ports 28 auto off speed 10000 duplex full
enable sharing 1 grouping 1 algorithm address-based L2 lacp
configure vlan Default add ports 2-3,6,8-15,17-18,20,22-36 untagged
configure vlan isc add ports 21 tagged
configure vlan tenant add ports 1,16,21 tagged
configure vlan trunk1_2 add ports 21 tagged
configure vlan trunk1_3 add ports 5 untagged
configure vlan trunk1_4 add ports 7 untagged
configure vlan untagtenant add ports 1,16,21 untagged
configure vlan isc ipaddress 172.16.1.0 255.255.255.254
configure vlan loop ipaddress 1.1.1.1 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.27 255.255.255.0
configure vlan mlagvtep ipaddress 1.1.1.200 255.255.255.255
enable ipforwarding vlan mlagvtep
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan trunk1_2 ipaddress 192.168.5.0 255.255.255.254
enable ipforwarding vlan trunk1_2
configure vlan trunk1_3 ipaddress 192.168.1.0 255.255.255.254
enable ipforwarding vlan trunk1_3
configure vlan trunk1_4 ipaddress 192.168.2.0 255.255.255.254
enable ipforwarding vlan trunk1_4
configure vlan untagtenant ipaddress 20.1.100.100 255.255.255.0
enable ipforwarding vlan untagtenant

#
# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"

#
# Module vsm configuration.
#
create mlag peer "leaf2"
```

```
configure mlag peer "leaf2" ipaddress 172.16.1.1 vr VR-Default
configure mlag peer "leaf2" lacp-mac 00:00:de:ad:be:ef
enable mlag port 1 peer "leaf2" id 1

#
# Module bgp configuration.
#
configure bgp AS-number 10000
configure bgp routerid 1.1.1.1
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 1.1.1.1/32
configure bgp add network 1.1.1.200/32
create bgp neighbor 192.168.1.1 remote-AS-number 20000
enable bgp neighbor 192.168.1.1
create bgp neighbor 192.168.2.1 remote-AS-number 20000
enable bgp neighbor 192.168.2.1
create bgp neighbor 192.168.5.1 remote-AS-number 10000
enable bgp neighbor 192.168.5.1
configure bgp neighbor 192.168.1.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.1.1 capability l2vpn-EVPN
configure bgp neighbor 192.168.2.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.2.1 capability l2vpn-EVPN
configure bgp neighbor 192.168.5.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.5.1 capability l2vpn-EVPN
enable bgp export direct address-family ipv4-unicast
enable bgp
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.1.1.200 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant
```

Leaf 2

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-72
configure vr VR-Default add ports 1-72
configure vlan default delete ports 1-72
enable jumbo-frame ports all
create vlan "isc"
configure vlan isc tag 4000
create vlan "loop"
enable loopback-mode vlan loop
create vlan "loop2"
enable loopback-mode vlan loop2
create vlan "mlagvtep"
enable loopback-mode vlan mlagvtep
create vlan "tenant"
```

```
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "trunk1_2"
configure vlan trunk1_2 tag 4001
create vlan "trunk2_3"
create vlan "trunk2_4"
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 50 auto off speed 10000 duplex full
configure ports 51 auto off speed 10000 duplex full
configure ports 52 auto off speed 10000 duplex full
enable sharing 53 grouping 53-56 algorithm address-based L2 lacp
configure vlan isc add ports 66 tagged
configure vlan tenant add ports 4-5,53,66 tagged
configure vlan trunk1_2 add ports 66 tagged
configure vlan trunk2_3 add ports 57 untagged
configure vlan trunk2_4 add ports 49 untagged
configure vlan untagtenant add ports 4-5,53,66 untagged
configure vlan isc ipaddress 172.16.1.1 255.255.255.254
configure vlan loop ipaddress 2.2.2.2 255.255.255.255
enable ipforwarding vlan loop
configure vlan loop2 ipaddress 2.2.2.100 255.255.255.255
enable ipforwarding vlan loop2
configure vlan Mgmt ipaddress 10.127.16.19 255.255.255.0
configure vlan mlagvtep ipaddress 1.1.1.200 255.0.0.0
enable ipforwarding vlan mlagvtep
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan trunk1_2 ipaddress 192.168.5.1 255.255.255.254
enable ipforwarding vlan trunk1_2
configure vlan trunk2_3 ipaddress 192.168.3.0 255.255.255.254
enable ipforwarding vlan trunk2_3
configure vlan trunk2_4 ipaddress 192.168.4.0 255.255.255.254
enable ipforwarding vlan trunk2_4
configure vlan untagtenant ipaddress 20.1.100.100 255.255.255.0
enable ipforwarding vlan untagtenant
#
# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"


#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.1.1.200 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant

#
# Module bgp configuration.
#
#
configure bgp AS-number 10000
configure bgp routerid 2.2.2.2
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 1.1.1.200/32
configure bgp add network 2.2.2.2/32
```

```
create bgp neighbor 192.168.3.1 remote-AS-number 20000
enable bgp neighbor 192.168.3.1
create bgp neighbor 192.168.4.1 remote-AS-number 20000
enable bgp neighbor 192.168.4.1
create bgp neighbor 192.168.5.0 remote-AS-number 10000
enable bgp neighbor 192.168.5.0
configure bgp neighbor 192.168.3.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.3.1 capability l2vpn-EVPN
configure bgp neighbor 192.168.4.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.4.1 capability l2vpn-EVPN
configure bgp neighbor 192.168.5.0 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.5.0 capability l2vpn-EVPN
enable bgp export direct address-family ipv4-unicast
enable bgp
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
# Module vrrp configuration.
#
create vrrp vlan tenant vrid 1
configure vrrp vlan tenant vrid 1 priority 255
create vrrp vlan untagtenant vrid 1
configure vrrp vlan untagtenant vrid 1 priority 255
configure vrrp vlan tenant vrid 1 add 10.1.100.100
configure vrrp vlan untagtenant vrid 1 add 20.1.100.100
enable vrrp vlan tenant vrid 1
enable vrrp vlan untagtenant vrid 1

#
# Module vsm configuration.
#
create mlag peer "leaf1"
configure mlag peer "leaf1" ipaddress 172.16.1.0 vr VR-Default
configure mlag peer "leaf1" lacp-mac 00:00:de:ad:be:ef
enable mlag port 53 peer "leaf1" id 1
```

Leaf 3

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-56
configure vr VR-Default add ports 1-56
configure vlan default delete ports 1,50-51
enable jumbo-frame ports all
create vlan "leaf3_trunk1"
create vlan "leaf3_trunk2"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "tenant"
configure vlan tenant tag 10
configure vlan tenant suppress arp-only
create vlan "untagtenant"
configure vlan untagtenant suppress arp-only
configure ports 49 auto off speed 10000 duplex full
configure ports 50 auto off speed 10000 duplex full
configure ports 51 auto off speed 10000 duplex full
configure ports 52 auto off speed 10000 duplex full
configure ports 54 auto off speed 10000 duplex full
configure ports 55 auto off speed 10000 duplex full
```

```
configure ports 56 auto off speed 10000 duplex full
configure vlan Default add ports 2-49,52-56 untagged
configure vlan leaf3_trunk1 add ports 51 untagged
configure vlan leaf3_trunk2 add ports 50 untagged
configure vlan tenant add ports 1 tagged
configure vlan untagtenant add ports 1 untagged
configure vlan leaf3_trunk1 ipaddress 192.168.6.0 255.255.255.254
enable ipforwarding vlan leaf3_trunk1
configure vlan leaf3_trunk2 ipaddress 192.168.7.0 255.255.255.254
enable ipforwarding vlan leaf3_trunk2
configure vlan loop ipaddress 5.5.5.5 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.17 255.255.255.0
configure vlan tenant ipaddress 10.1.100.100 255.255.255.0
enable ipforwarding vlan tenant
configure vlan untagtenant ipaddress

# Module mcmgr configuration.
#
disable igmp snooping vlan "tenant"
disable igmp snooping vlan "untagtenant"

#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 5.5.5.5 vr "VR-Default"
create virtual-network "vni1" flooding standard
configure virtual-network "vni1" VXLAN vni 100
configure virtual-network "vni1" add vlan tenant
create virtual-network "vni2" flooding standard
configure virtual-network "vni2" VXLAN vni 200
configure virtual-network "vni2" add vlan untagtenant

#
# Module bgp configuration.
#
configure bgp AS-number 30000
configure bgp routerid 5.5.5.5
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp restart both
configure bgp add network 5.5.5.5/32
create bgp neighbor 192.168.6.1 remote-AS-number 20000
enable bgp neighbor 192.168.6.1
create bgp neighbor 192.168.7.1 remote-AS-number 20000
enable bgp neighbor 192.168.7.1
configure bgp neighbor 192.168.6.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.6.1 capability l2vpn-EVPN
configure bgp neighbor 192.168.7.1 allowas-in max-as-occurrence 3
enable bgp neighbor 192.168.7.1 capability l2vpn-EVPN
enable bgp
create bgp EVPN instance evi_1
configure bgp EVPN instance evi_1 VXLAN vni 100
create bgp EVPN instance evi_2
configure bgp EVPN instance evi_2 VXLAN vni 200
configure bgp EVPN instance evi_1 route-target both add 100:100
configure bgp EVPN instance evi_2 route-target both add 200:200
#
# Module vrrp configuration.
#
create vrrp vlan tenant vrid 1
configure vrrp vlan tenant vrid 1 priority 255
create vrrp vlan untagtenant vrid 1
configure vrrp vlan untagtenant vrid 1 priority 255
```

```
configure vrrp vlan tenant vrid 1 add 10.1.100.100
configure vrrp vlan untagtenant vrid 1 add 20.1.100.100
enable vrrp vlan tenant vrid 1
enable vrrp vlan untagtenant vrid 1
```

Spine 1

```
configure vlan default delete ports all
configure vr VR-Default delete ports 1-128
configure vr VR-Default add ports 1-128
configure vlan default delete ports 33,57,89,105
enable jumbo-frame ports all
create vlan "leaf3_trunk1"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "routed"
configure vlan routed tag 4030
create vlan "trunk1_3"
create vlan "trunk2_3"
configure vlan Default add ports 1-32,34-56,58-88,90-104,106-128 untagged
configure vlan leaf3_trunk1 add ports 89 untagged
configure vlan routed add ports 33 tagged
configure vlan trunk1_3 add ports 57 untagged
configure vlan trunk2_3 add ports 105 untagged
configure vlan leaf3_trunk1 ipaddress 192.168.6.1 255.255.255.254
enable ipforwarding vlan leaf3_trunk1
configure vlan loop ipaddress 3.3.3.100 255.0.0.0
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.16.24 255.255.255.0
configure vlan routed ipaddress 172.16.1.1 255.255.255.0
enable ipforwarding vlan routed
configure vlan trunk1_3 ipaddress 192.168.1.1 255.255.255.254
enable ipforwarding vlan trunk1_3
configure vlan trunk2_3 ipaddress 192.168.3.1 255.255.255.254
enable ipforwarding vlan trunk2_3

#
# Module bgp configuration.
#
#
configure bgp AS-number 20000
configure bgp routerid 3.3.3.3
configure bgp maximum-paths 2
enable bgp fast-external-fallover
create bgp neighbor 192.168.1.0 remote-AS-number 10000
enable bgp neighbor 192.168.1.0
create bgp neighbor 192.168.3.0 remote-AS-number 10000
enable bgp neighbor 192.168.3.0
create bgp neighbor 192.168.6.0 remote-AS-number 30000
enable bgp neighbor 192.168.6.0
enable bgp neighbor 192.168.1.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.1.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.3.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.3.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.6.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.6.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp export ospf-extern1 address-family ipv4-unicast
enable bgp export ospf-extern2 address-family ipv4-unicast
enable bgp export ospf-inter address-family ipv4-unicast
enable bgp export ospf-intra address-family ipv4-unicast
enable bgp
# Module ospf configuration.
#
```

```
enable ospf
configure ospf add vlan routed area 0.0.0.0
```

Spine 2

```
#
# Module vlan configuration.
#
configure vlan default delete ports all
configure vr VR-Default delete ports 1-64
configure vr VR-Default add ports 1-64
configure vlan default delete ports 25,27,49
create vlan "leaf3_trunk2"
create vlan "loop"
enable loopback-mode vlan loop
create vlan "trunk1_4"
create vlan "trunk2_4"
configure ports 1 auto off speed 10000 duplex full
configure ports 2 auto off speed 10000 duplex full
configure ports 3 auto off speed 10000 duplex full
configure ports 4 auto off speed 10000 duplex full
configure ports 5 auto off speed 10000 duplex full
configure ports 6 auto off speed 10000 duplex full
configure ports 7 auto off speed 10000 duplex full
configure ports 8 auto off speed 10000 duplex full
configure ports 9 auto off speed 10000 duplex full
configure ports 10 auto off speed 10000 duplex full
configure ports 11 auto off speed 10000 duplex full
configure ports 12 auto off speed 10000 duplex full
configure ports 13 auto off speed 10000 duplex full
configure ports 14 auto off speed 10000 duplex full
configure ports 15 auto off speed 10000 duplex full
configure ports 16 auto off speed 10000 duplex full
configure ports 17 auto off speed 10000 duplex full
configure ports 18 auto off speed 10000 duplex full
configure ports 19 auto off speed 10000 duplex full
configure ports 20 auto off speed 10000 duplex full
configure ports 21 auto off speed 10000 duplex full
configure ports 22 auto off speed 10000 duplex full
configure ports 23 auto off speed 10000 duplex full
configure ports 24 auto off speed 10000 duplex full
configure ports 25 auto off speed 10000 duplex full
configure ports 26 auto off speed 10000 duplex full
configure ports 27 auto off speed 10000 duplex full
configure ports 28 auto off speed 10000 duplex full
configure ports 29 auto off speed 10000 duplex full
configure ports 30 auto off speed 10000 duplex full
configure ports 31 auto off speed 10000 duplex full
configure ports 32 auto off speed 10000 duplex full
configure ports 33 auto off speed 10000 duplex full
configure ports 34 auto off speed 10000 duplex full
configure ports 35 auto off speed 10000 duplex full
configure ports 36 auto off speed 10000 duplex full
configure ports 37 auto off speed 10000 duplex full
configure ports 38 auto off speed 10000 duplex full
configure ports 39 auto off speed 10000 duplex full
configure ports 40 auto off speed 10000 duplex full
configure ports 41 auto off speed 10000 duplex full
configure ports 42 auto off speed 10000 duplex full
configure ports 43 auto off speed 10000 duplex full
configure ports 44 auto off speed 10000 duplex full
configure ports 45 auto off speed 10000 duplex full
configure ports 46 auto off speed 10000 duplex full
configure ports 47 auto off speed 10000 duplex full
```

```
configure ports 48 auto off speed 10000 duplex full
configure vlan Default add ports 1-24,26,28-48,50-64 untagged
configure vlan leaf3_trunk2 add ports 25 untagged
configure vlan trunk1_4 add ports 27 untagged
configure vlan trunk2_4 add ports 49 untagged
configure vlan leaf3_trunk2 ipaddress 192.168.7.1 255.255.255.254
enable ipforwarding vlan leaf3_trunk2
configure vlan loop ipaddress 4.4.4.4 255.255.255.255
enable ipforwarding vlan loop
configure vlan Mgmt ipaddress 10.127.13.10 255.255.255.0
configure vlan trunk1_4 ipaddress 192.168.2.1 255.255.255.254
enable ipforwarding vlan trunk1_4
configure vlan trunk2_4 ipaddress 192.168.4.1 255.255.255.254
enable ipforwarding vlan trunk2_4
#
# Module bgp configuration.
#
configure bgp AS-number 20000
configure bgp routerid 4.4.4.4
configure bgp maximum-paths 2
enable bgp fast-external-fallover
configure bgp add network 3.3.3.100/32
configure bgp add network 4.4.4.4/32
create bgp neighbor 192.168.2.0 remote-AS-number 10000
enable bgp neighbor 192.168.2.0
create bgp neighbor 192.168.4.0 remote-AS-number 10000
enable bgp neighbor 192.168.4.0
create bgp neighbor 192.168.7.0 remote-AS-number 30000
enable bgp neighbor 192.168.7.0
enable bgp neighbor 192.168.2.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.2.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.4.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.4.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp neighbor 192.168.7.0 capability l2vpn-EVPN
enable bgp neighbor 192.168.7.0 address-family l2vpn-EVPN next-hop-unchanged
enable bgp export ospf-extern1 address-family ipv4-unicast
enable bgp export ospf-extern2 address-family ipv4-unicast
enable bgp
```

Verification

Verify BGP Adjacency

```
Leaf1.21 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.1.1                             20000      1      ESTABLISHED  45     62
(0    ) 0:0:08:40
Ee-- 192.168.2.1                             20000      1      ESTABLISHED  27     74
(0    ) 0:0:08:50
Ie-- 192.168.5.1                             10000      1      ESTABLISHED  57     60
(0    ) 0:0:08:35

Leaf2.10 # show bgp neighbor

     Peer                                    AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.3.1                             20000      1      ESTABLISHED  75     64
(0    ) 0:0:09:08
```

```
Ee-- 192.168.4.1                            20000      1      ESTABLISHED  64     77
(0    ) 0:0:09:01
Ie-- 192.168.5.0                            10000      1      ESTABLISHED  61     58
(0    ) 0:0:09:04

Leaf3.11 # show bgp neighbor

     Peer                                   AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.6.1                            20000      1      ESTABLISHED  57     36
(0    ) 0:0:09:22
Ee-- 192.168.7.1                            20000      1      ESTABLISHED  50     73
(0    ) 0:0:09:14

Spine1.2 # show bgp neighbor

     Peer                                   AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.1.0                            10000      1      ESTABLISHED  63     46
(0    ) 0:0:09:36
Ee-- 192.168.3.0                            10000      1      ESTABLISHED  64     76
(0    ) 0:0:09:35
Ee-- 192.168.6.0                            30000      1      ESTABLISHED  37     57
(0    ) 0:0:09:35

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 3                IBGP Peers          : 0
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0

Spine2.14 # show bgp neighbor

     Peer                                   AS         Weight State        InMsgs
OutMsgs(InQ)  Up/Down
-------------------------------------------------------------------------------------------
------------------
Ee-- 192.168.2.0                            10000      1      ESTABLISHED  234    186
(0    ) 0:0:10:00
Ee-- 192.168.4.0                            10000      1      ESTABLISHED  218    244
(0    ) 0:0:09:42
Ee-- 192.168.7.0                            30000      1      ESTABLISHED  146    134
(0    ) 0:0:09:41

Flags: (d) disabled, (e) enabled, (E) external peer, (I) internal peer
       (m) EBGP multihop, (r) route reflector client

BGP Peer Statistics
  Total Peers       : 3
  EBGP Peers        : 3                IBGP Peers          : 0
  RR Client         : 0                EBGP Multihop       : 0
  Enabled           : 3                Disabled            : 0
```

Verify Type 2 MAC Routes

```
Leaf1.5 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
```

```
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 00:00:0b:cb:21:a4
100                                          Yes
L        10 00:00:0c:14:88:4b
100                                          Yes
L        10 00:00:0c:34:c1:ed
100                                          Yes
L        10 00:04:96:9d:42:32
100                                          Yes
L        10 00:04:96:a3:fb:b0
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4087 00:00:00:1c:7e:5f
200                                          Yes
R      4087 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4087 00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4087 00:00:04:d6:4d:6e
200                                          Yes
L      4087 00:00:04:d6:4d:6f
200                                          Yes
L      4087 00:04:96:9d:42:32
200                                          Yes
L      4087 00:04:96:a3:fb:b0
200                                          Yes
R      4087 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 16

Leaf2.5 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 00:00:0b:cb:21:a4
100                                          Yes
L        10 00:00:0c:14:88:4b
100                                          Yes
L        10 00:00:0c:34:c1:ed
100                                          Yes
L        10 00:04:96:9d:42:32
100                                          Yes
L        10 00:04:96:a3:fb:b0
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4088 00:00:00:1c:7e:5f
200                                          Yes
```

```
R      4088 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4088 00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4088 00:00:04:d6:4d:6e
200                                          Yes
L      4088 00:00:04:d6:4d:6f
200                                          Yes
L      4088 00:04:96:9d:42:32
200                                          Yes
L      4088 00:04:96:a3:fb:b0
200                                          Yes
R      4088 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 16
Leaf3.12 #  show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
R        10 00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
L        10 00:00:00:47:ce:72
100                                          Yes
R        10 00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
L        10 00:11:88:ff:04:74
100                                          Yes
R      4090 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4090 00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
L      4090 00:00:00:47:ce:73
200                                          Yes
R      4090 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4090 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4090 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4090 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
```

```
L      4090 00:11:88:ff:04:74
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 21

Spine1.7 # show bgp EVPN mac
Src EVI-Idx MAC               BGP Next Hop           VNI ESI
ESI-Port   In Use Sequence
--- ------- ----------------- --------------- ---------- ------------------------------
---------- ------ --------
L        10 00:00:00:1c:7e:60
100                                          Yes
R        10 00:00:00:47:ce:72 5.5.5.5
100                                          Yes
R        10 00:00:0b:cb:21:a4 1.1.1.200
100                                          Yes
R        10 00:00:0c:14:88:4b 1.1.1.200
100                                          Yes
R        10 00:00:0c:34:c1:ed 1.1.1.200
100                                          Yes
R        10 00:04:96:9d:42:32 1.1.1.200
100                                          Yes
R        10 00:04:96:a3:fb:b0 1.1.1.200
100                                          Yes
R        10 00:11:88:ff:04:74 5.5.5.5
100                                          Yes
R      4091 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
R      4091 00:00:00:1c:7e:5f 1.1.1.200
200                                          Yes
L      4091 00:00:00:1c:7e:61
200                                          Yes
R      4091 00:00:00:47:ce:73 5.5.5.5
200                                          Yes
R      4091 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6e 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 00:00:04:d6:4d:6f 1.1.1.200
200                                          Yes
R      4091 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 00:04:96:9d:42:32 1.1.1.200
200                                          Yes
R      4091 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 00:04:96:a3:fb:b0 1.1.1.200
200                                          Yes
R      4091 00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in MAC forwarding database.

Total MAC/IP entries displayed: 21
```

Verify Type 2 IPv4 Routes

```
Leaf1.6 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
----------------------------- ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 10.1.100.6      00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 10.1.100.1      00:00:0b:cb:21:a4
100                                          Yes
L        10 10.1.100.5      00:00:0c:14:88:4b
100                                          Yes
L        10 10.1.100.100    00:04:96:a3:fb:b0
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4087 20.1.100.5      00:00:00:1c:7e:5f
200                                          Yes
R      4087 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4087 20.1.100.6      00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4087 20.1.100.1      00:00:04:d6:4d:6e
200                                          Yes
L      4087 20.1.100.2      00:00:04:d6:4d:6f
200                                          Yes
L      4087 20.1.100.100    00:04:96:a3:fb:b0
200                                          Yes
R      4087 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                          Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 13

Leaf2.6 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                            ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
----------------------------- ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                          Yes
R        10 10.1.100.6      00:00:00:47:ce:72 5.5.5.5
100                                          Yes
L        10 10.1.100.1      00:00:0b:cb:21:a4
100                                          Yes
L        10 10.1.100.5      00:00:0c:14:88:4b
100                                          Yes
L        10 10.1.100.2      00:00:0c:34:c1:ed
100                                          Yes
L        10 10.1.100.100    00:04:96:9d:42:32
100                                          Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                          Yes
L      4088 20.1.100.5      00:00:00:1c:7e:5f
200                                          Yes
R      4088 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                          Yes
R      4088 20.1.100.6      00:00:00:47:ce:73 5.5.5.5
200                                          Yes
L      4088 20.1.100.1      00:00:04:d6:4d:6e
200                                          Yes
```

```
L      4088 20.1.100.2      00:00:04:d6:4d:6f
200                                        Yes
L      4088 20.1.100.100    00:04:96:9d:42:32
200                                        Yes
R      4088 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                        Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 14

Leaf3.13 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                           ESI-Port   In Use
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
R        10 10.1.100.3      00:00:00:1c:7e:60 3.3.3.100
100                                        Yes
L        10 10.1.100.6      00:00:00:47:ce:72
100                                        Yes
R        10 10.1.100.1      00:00:0b:cb:21:a4 1.1.1.200
100                                        Yes
R        10 10.1.100.5      00:00:0c:14:88:4b 1.1.1.200
100                                        Yes
R        10 10.1.100.2      00:00:0c:34:c1:ed 1.1.1.200
100                                        Yes
R        10 10.1.100.100    00:04:96:9d:42:32 1.1.1.200
100                                        Yes
R        10 10.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
100                                        Yes
L        10 10.1.100.100    00:11:88:ff:04:74
100                                        Yes
R      4090 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                        Yes
R      4090 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                        Yes
R      4090 20.1.100.3      00:00:00:1c:7e:61 3.3.3.100
200                                        Yes
L      4090 20.1.100.6      00:00:00:47:ce:73
200                                        Yes
R      4090 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                        Yes
R      4090 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                        Yes
R      4090 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                        Yes
R      4090 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                        Yes
R      4090 20.1.100.100    00:04:96:9d:42:32 1.1.1.200
200                                        Yes
R      4090 20.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
200                                        Yes
L      4090 20.1.100.100    00:11:88:ff:04:74
200                                        Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 19

Spine1.13 # show bgp EVPN ipv4
Src EVI-Idx Destination     MAC               BGP Next Hop           VNI
ESI                           ESI-Port   In Use
```

```
--- ------- --------------- ----------------- --------------- ----------
------------------------------ ---------- ------
L        10 10.1.100.3      00:00:00:1c:7e:60
100                                        Yes
R        10 10.1.100.6      00:00:00:47:ce:72 5.5.5.5
100                                        Yes
R        10 10.1.100.1      00:00:0b:cb:21:a4 1.1.1.200
100                                        Yes
R        10 10.1.100.5      00:00:0c:14:88:4b 1.1.1.200
100                                        Yes
R        10 10.1.100.2      00:00:0c:34:c1:ed 1.1.1.200
100                                        Yes
R        10 10.1.100.100    00:04:96:9d:42:32 1.1.1.200
100                                        Yes
R        10 10.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
100                                        Yes
R        10 10.1.100.100    00:11:88:ff:04:74 5.5.5.5
100                                        Yes
R      4091 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                        Yes
R      4091 20.1.100.5      00:00:00:1c:7e:5f 1.1.1.200
200                                        Yes
L      4091 20.1.100.3      00:00:00:1c:7e:61
200                                        Yes
R      4091 20.1.100.6      00:00:00:47:ce:73 5.5.5.5
200                                        Yes
R      4091 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                        Yes
R      4091 20.1.100.1      00:00:04:d6:4d:6e 1.1.1.200
200                                        Yes
R      4091 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                        Yes
R      4091 20.1.100.2      00:00:04:d6:4d:6f 1.1.1.200
200                                        Yes
R      4091 20.1.100.100    00:04:96:9d:42:32 1.1.1.200
200                                        Yes
R      4091 20.1.100.100    00:04:96:a3:fb:b0 1.1.1.200
200                                        Yes
R      4091 20.1.100.100    00:11:88:ff:04:74 5.5.5.5
200                                        Yes

Src: (L) Local, (R) Remote
In Use: Yes/No - Indicates if entry is installed in IP ARP cache.

Total MAC/IP entries displayed: 19
```

Verify ARP and FDB Entries

```
Leaf1.7 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    10.1.100.1       00:00:0b:cb:21:a4   17      NO  tenant        10    16
VR-Default    10.1.100.5       00:00:0c:14:88:4b    7      NO  tenant        10    1
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    192.168.1.1      00:04:96:9c:2d:24   20      NO  trunk1_3      4089  5
VR-Default    192.168.2.1      00:04:96:9b:b7:c9   20      NO  trunk1_4      4088  7
VR-Default    20.1.100.1       00:00:04:d6:4d:6e   12      NO  untagtenant   4087  16
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    7      NO  untagtenant   4087  21
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    7      NO  untagtenant   4087  1
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4087  VR-
Default:5.5.5.5
VR-Default    192.168.5.1      00:04:96:9d:42:32   20      NO  trunk1_2      4001  21
VR-Default    172.16.1.1       00:04:96:9d:42:32   20      NO  isc           4000  21
```

```
Dynamic Entries  :           9             Static Entries           :          2
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh              :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning  :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

Leaf1.8 #  show fdb
MAC                                    VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                    untagtenant(4087) 0000  d mi       S   1
00:00:00:47:ce:72                         tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:00:47:ce:73                    untagtenant(4087) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:04:d6:4d:6e                    untagtenant(4087) 0000  d mi           16
00:00:04:d6:4d:6f                    untagtenant(4087) 0000  d mi       S   21
00:00:0b:cb:21:a4                         tenant(0010) 0000  d mi           16
00:00:0c:14:88:4b                         tenant(0010) 0000  d mi       S   1
00:00:0c:34:c1:ed                         tenant(0010) 0078  d m        S   21
00:04:96:9b:b7:c9                       trunk1_4(4088) 0000  d mi           7
00:04:96:9c:2d:24                       trunk1_3(4089) 0000  d mi           5
00:04:96:9d:42:32                         tenant(0010) 0000  s m        S   21
00:04:96:9d:42:32                    untagtenant(4087) 0000  s m        S   21
00:04:96:9d:42:32                       trunk1_2(4001) 0000  s m        S   21
00:04:96:9d:42:32                            isc(4000) 0000  s m        S   21
00:11:88:ff:04:74                         tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:11:88:ff:04:74                    untagtenant(4087) 0000  s m        SXE VR-
Default:5.5.5.5

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 16 Static: 8  Perm: 0  Dyn: 8  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300

Leaf2.7 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    7      NO  tenant        10    66
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    7      NO  tenant        10    4
VR-Default    10.1.100.5       00:00:0c:14:88:4b   31      NO  tenant        10    53
VR-Default    10.1.100.6       00:00:00:47:ce:72    0     YES  tenant        10    VR-
Default:5.5.5.5
VR-Default    192.168.3.1      00:04:96:9c:2d:24   33      NO  trunk2_3      4090  57
VR-Default    192.168.4.1      00:04:96:9b:b7:c9   33      NO  trunk2_4      4089  49
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    7      NO  untagtenant   4088  66
VR-Default    20.1.100.2       00:00:04:d6:4d:6f   31      NO  untagtenant   4088  4
VR-Default    20.1.100.5       00:00:00:1c:7e:5f   31      NO  untagtenant   4088  53
```

```
VR-Default    20.1.100.6       00:00:00:47:ce:73    0     YES  untagtenant   4088  VR-
Default:5.5.5.5
VR-Default    192.168.5.0      00:04:96:a3:fb:b0   33      NO  trunk1_2      4001  66
VR-Default    172.16.1.0       00:04:96:a3:fb:b0   33      NO  isc           4000  66

Dynamic Entries  :          10             Static Entries            :          2
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          60 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :     2700000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

(Software Update Required) Leaf2.8 # show fdb
MAC                                   VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                    untagtenant(4088) 0000  d mi       S   53
00:00:00:47:ce:72                         tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:00:47:ce:73                    untagtenant(4088) 0000  s m        SXE VR-
Default:5.5.5.5
00:00:04:d6:4d:6e                    untagtenant(4088) 0000  d mi       S   66
00:00:04:d6:4d:6f                    untagtenant(4088) 0000  d mi           4
00:00:0b:cb:21:a4                         tenant(0010) 0000  d mi       S   66
00:00:0c:14:88:4b                         tenant(0010) 0000  d mi       S   53
00:00:0c:34:c1:ed                         tenant(0010) 0000  d mi           4
00:04:96:9b:b7:c9                       trunk2_4(4089) 0000  d mi           49
00:04:96:9c:2d:24                       trunk2_3(4090) 0000  d mi           57
00:04:96:a3:fb:b0                         tenant(0010) 0000  s m        S   66
00:04:96:a3:fb:b0                    untagtenant(4088) 0000  s m        S   66
00:04:96:a3:fb:b0                       trunk1_2(4001) 0000  s m        S   66
00:04:96:a3:fb:b0                            isc(4000) 0000  s m        S   66
00:11:88:ff:04:74                         tenant(0010) 0000  s m        SXE VR-
Default:5.5.5.5
00:11:88:ff:04:74                    untagtenant(4088) 0000  s m        SXE VR-
Default:5.5.5.5

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 16 Static: 8  Perm: 0  Dyn: 8  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300

Leaf3.14 # show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    192.168.6.1      00:04:96:9c:2d:24   20      NO  leaf3_trunk1  4094  51
VR-Default    192.168.7.1      00:04:96:9b:b7:c9   20      NO  leaf3_trunk2  4093  50
VR-Default    10.1.100.1       00:00:0b:cb:21:a4    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.2       00:00:0c:34:c1:ed    0     YES  tenant        10    VR-
```

```
Default:1.1.1.200
VR-Default    10.1.100.5       00:00:0c:14:88:4b    0     YES  tenant        10    VR-
Default:1.1.1.200
VR-Default    10.1.100.6       00:00:00:47:ce:72   15      NO  tenant        10    1
VR-Default    20.1.100.1       00:00:04:d6:4d:6e    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.2       00:00:04:d6:4d:6f    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.5       00:00:00:1c:7e:5f    0     YES  untagtenant   4090  VR-
Default:1.1.1.200
VR-Default    20.1.100.6       00:00:00:47:ce:73   20      NO  untagtenant   4090  1

Dynamic Entries  :           4             Static Entries            :          6
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :         Off

ARP Global Settings
Max Entries         :    12288
Max Pending Entries :      256
Max Proxy Entries   :      256

(Software Update Required) Leaf3.15 # show fdb
MAC                                     VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:00:00:1c:7e:5f                     untagtenant(4090) 0000  s mi        XE VR-
Default:1.1.1.200
00:00:00:47:ce:72                          tenant(0010) 0000  d mi           1
00:00:00:47:ce:73                     untagtenant(4090) 0000  d mi           1
00:00:04:d6:4d:6e                     untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:00:04:d6:4d:6f                     untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0b:cb:21:a4                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:00:0c:14:88:4b                          tenant(0010) 0000  s mi        XE VR-
Default:1.1.1.200
00:00:0c:34:c1:ed                          tenant(0010) 0000  s mi        XE VR-
Default:1.1.1.200
00:04:96:9b:b7:c9                    leaf3_trunk2(4093) 0000  d mi           50
00:04:96:9c:2d:24                    leaf3_trunk1(4094) 0000  d mi           51
00:04:96:9d:42:32                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:9d:42:32                     untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                          tenant(0010) 0000  s m         XE VR-
Default:1.1.1.200
00:04:96:a3:fb:b0                     untagtenant(4090) 0000  s m         XE VR-
Default:1.1.1.200

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN
```

```
Total: 14 Static: 10  Perm: 0  Dyn: 4  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
```

EVPN with Symetric Routing Configuration Example

Configuration Notes

The topology is: Leaf1----ebgp-----Spine----ebgp----Leaf2.

The configurations show the BGP, OTM, and VLAN configuration with static and direct routes redistributed into EVPN at each leaf and advertised to the other leaf node.

OSPF runs between each router to provide routes to each loopback, which are tunnel endpoints.

Leaf 1 Configuration

```
#
# Module devmgr configuration.
#
configure snmp sysContact "support@extremenetworks.com, +1 888 257 3000"
configure sys-recovery-level switch reset
#
# Module vlan configuration.
#
create vr "vr1" type vrf vr "VR-Default"
create vlan "loop1"
enable loopback-mode vlan loop1
create vlan "loopvr1" vr vr1
enable loopback-mode vlan loopvr1
create vlan "vlan48" vr vr1
configure vlan vlan48 tag 48
configure vlan vlan48 add ports 48 tagged
configure vlan loop1 ipaddress 1.0.0.40 255.255.255.255
enable ipforwarding vlan loop1
configure vlan loopvr1new ipaddress 1.0.0.40 255.255.255.255
enable ipforwarding vlan loopvr1
configure vlan vlan48 ipaddress 192.168.48.1 255.255.255.0
enable ipforwarding vlan vlan48

#
# Module mcmgr configuration.
#
disable igmp snooping vlan "loopvr1new"
disable igmp snooping vlan "vlan48"

#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.0.0.40 vr "VR-Default"
create virtual-network "vnetvr1" flooding standard
configure virtual-network "vnetvr1" vxlan vni 123
configure virtual-network "vnetvr1" add vlan loopvr1new
create virtual-network "vr1_routing" flooding standard
configure virtual-network "vr1_routing" vxlan vni 678
configure virtual-network "vr1_routing" add vlan vlan48
#
# Module fdb configuration.
#
```

```
#
# Module rtmgr configuration.
#

configure iproute add default 10.68.13.1 vr VR-Mgmt
configure iproute add 3.0.0.0 255.0.0.0 192.168.48.2 vr vr1
configure iproute add 2.0.0.0 255.0.0.0 192.168.48.2 vr vr1
configure iproute add 1.0.0.0 255.0.0.0 192.168.48.2 vr vr1


#
# Module bgp configuration.
#
configure bgp AS-number 40
configure bgp routerid 1.0.0.40
configure bgp add network 1.0.0.40/32
create bgp neighbor 192.168.99.2 remote-AS-number 38
enable bgp neighbor 192.168.99.2
enable bgp neighbor 192.168.99.2 capability l2vpn-evpn
enable bgp export direct address-family l2vpn-evpn vr vr1
enable bgp export static address-family l2vpn-evpn vr vr1
configure bgp evpn l3vni 123 vr vr1
enable bgp

#
# Module ospf configuration.
#
enable ospf
enable ospf export direct cost 0 type ase-type-1
configure ospf add vlan vlan99 area 0.0.0.0
```

Leaf 2 Configuration

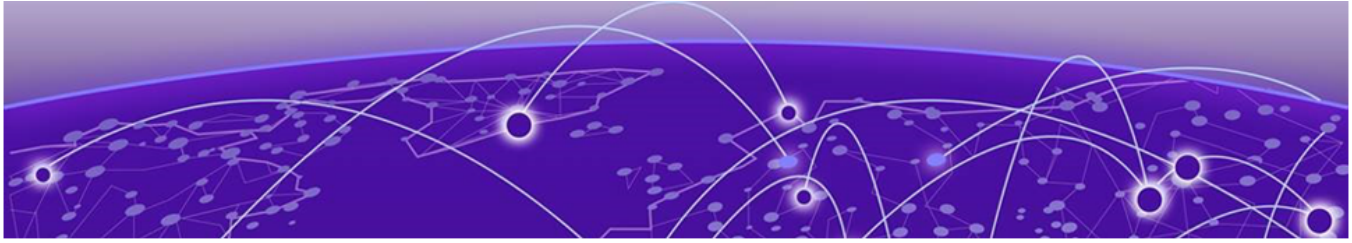
```
#
# Module vlan configuration.
#

configure vr VR-Default add ports 1,3-7,9-47,49,51-72
create vr "vr1" type vrf vr "VR-Default"
create vlan "loop1"
enable loopback-mode vlan loop1
create vlan "loopvr1" vr vr1
enable loopback-mode vlan loopvr1
create vlan "vlan48" vr vr1
configure vlan vlan48 tag 48
create vlan "vlan66"
enable ipforwarding ipv6 vlan vlan66
configure vlan vlan66 tag 66
configure ports 1 auto off speed 10000 duplex full
configure vlan vlan48 add ports 48 tagged
configure vlan vlan66 add ports 2 tagged
configure vlan loop1 ipaddress 1.0.0.25 255.255.255.255
enable ipforwarding vlan loop1
configure vlan loopvr1 ipaddress 1.0.0.25 255.255.255.255
enable ipforwarding vlan loopvr1
configure vlan vlan48 ipaddress 192.168.48.11 255.255.255.0
enable ipforwarding vlan vlan48
configure vlan vlan66 ipaddress 192.168.66.1 255.255.255.0
enable ipforwarding vlan vlan66

#
# Module mcmgr configuration.
#
disable igmp snooping vlan "loopvr1"
```

```
disable igmp snooping vlan "vlan48"
#
# Module otm configuration.
#
configure virtual-network local-endpoint ipaddress 1.0.0.25 vr "VR-Default"
create virtual-network "vnetvr1" flooding standard
configure virtual-network "vnetvr1" vxlan vni 123
configure virtual-network "vnetvr1" add vlan loopvr1new
create virtual-network "vr1_routing" flooding standard
configure virtual-network "vr1_routing" vxlan vni 679
configure virtual-network "vr1_routing" add vlan vlan48

#
# Module fdb configuration.
#
#
# Module rtmgr configuration.
#
configure iproute add 103.0.0.0 255.0.0.0 192.168.48.99 vr vr1
configure iproute add 102.0.0.0 255.0.0.0 192.168.48.99 vr vr1
configure iproute add 101.0.0.0 255.0.0.0 192.168.48.99 vr vr1
#
# Module bgp configuration.
#
configure bgp AS-number 25
configure bgp routerid 1.0.0.25
configure bgp add network 1.0.0.25/32
create bgp neighbor 192.168.66.2 remote-AS-number 38
enable bgp neighbor 192.168.66.2
enable bgp neighbor 192.168.66.2 capability l2vpn-evpn
enable bgp export direct address-family l2vpn-evpn vr vr1
enable bgp export static address-family l2vpn-evpn vr vr1
configure bgp evpn l3vni 123 vr vr1
enable bgp
#
# Module ospf configuration.
#
enable ospf
enable ospf export direct cost 0 type ase-type-1
configure ospf add vlan vlan66 area 0.0.0.0
```

Spine Configuration

```
#
# Module vlan configuration.
#
create vlan "loop1"
enable loopback-mode vlan loop1
create vlan "vlan99"
configure vlan vlan99 tag 99
create vlan "vlan66"
configure vlan vlan66 tag 66
configure vlan vlan99 add ports 2 tagged
configure vlan vlan66 add ports 1 tagged
configure vlan loop1 ipaddress 1.0.0.38 255.255.255.255
enable ipforwarding vlan loop1
configure vlan vlan99 ipaddress 192.168.99.2 255.255.255.0
enable ipforwarding vlan vlan99
configure vlan vlan66 ipaddress 192.168.66.2 255.255.255.0
enable ipforwarding vlan vlan66
#
# Module bgp configuration.
#
configure bgp AS-number 38
```

```
configure bgp routerid 1.0.0.38
configure bgp maximum-paths 2
enable bgp multipath-relax
create bgp neighbor 192.168.66.1 remote-AS-number 25
enable bgp neighbor 192.168.66.1
create bgp neighbor 192.168.99.1 remote-AS-number 40
enable bgp neighbor 192.168.99.1
enable bgp neighbor 192.168.66.1 capability l2vpn-evpn
enable bgp neighbor 192.168.66.1 address-family l2vpn-evpn next-hop-unchanged
enable bgp neighbor 192.168.99.1 capability l2vpn-evpn
enable bgp neighbor 192.168.99.1 address-family l2vpn-evpn next-hop-unchanged
enable bgp
#
# Module ospf configuration.
#
enable ospf
enable ospf export direct cost 0 type ase-type-1
configure ospf add vlan vlan99 area 0.0.0.0
configure ospf add vlan vlan66 area 0.0.0.0
```



Extended Edge Switching

Extended Edge Switching Overview on page 734
Supported Bridge Port Extender Models on page 748
Supported Platforms on page 749
Extended Edge Switching on Stacking Overview on page 749
Bridge Port Extender General Limitations on page 752
ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders on page 753
Using VPEX Bridge Port Extenders on page 757
Configuration Examples on page 770
Upgrading the Controlling Bridge and Bridge Ports Extenders on page 782

Extended Edge Switching Overview

What Is Extended Edge Switching?

Virtual Port Extenders (VPEX) Description

Figure 69 on page 735 shows the VPEX switching architecture, based on the *IEEE 802.1BR* specification, comprising one or two controlling bridges (CBs), and one or more bridge port extenders (BPEs) (see Bridge Port Extenders Description on page 735). In this document, BPEs are V400 Virtual Port Extenders, and CBs are ExtremeXOS switches.

In this architecture, ports on the CB or BPEs connecting to BPEs are termed *cascade ports*, while corresponding ports on BPEs connecting them to the CB or upstream BPEs are termed *upstream ports*. Ports from the BPEs connected to the rest of the network are termed *extended ports*.

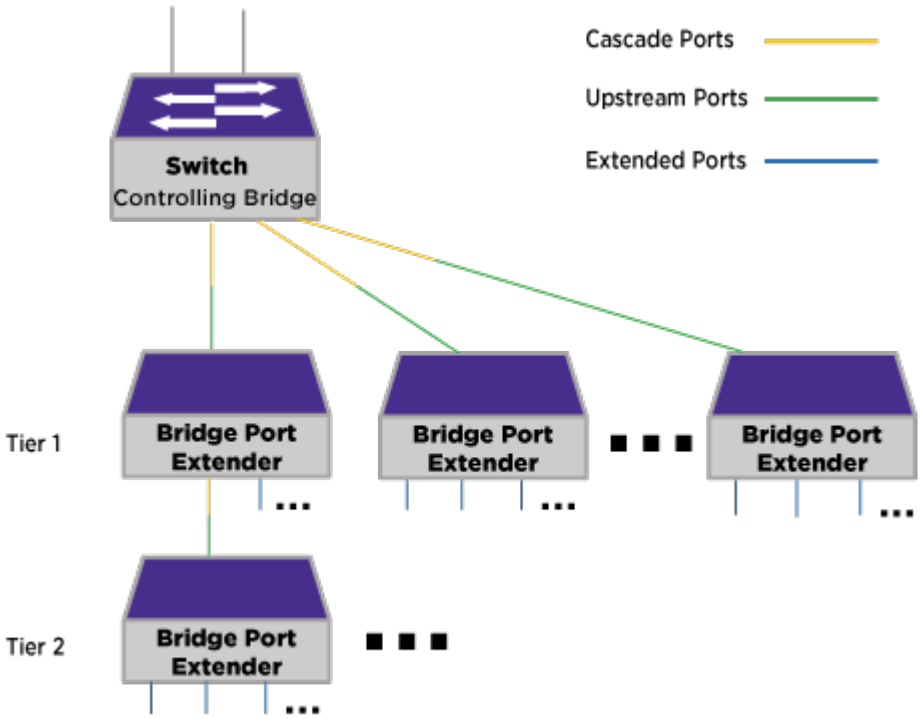


Figure 69: VPEX Architecture

Bridge Port Extenders Description

Bridge port extenders (BPEs) are devices that do not fully process packets, nor make forwarding or filtering decisions. Instead, BPEs simply receive packets from extended ports and forward packets towards the upstream controlling bridge for L2/L3/Policy processing.

Note
Due to the lack of extensive processing performed on BPEs, traffic entering on a BPE extended port that is destined to exit another extended port on the same BPE is forwarded to the controlling bridge (CB), and then forwarded back to the BPE, rather than processed exclusively within the BPE.

Important Terminology

The following terms are important for understanding virtual port extenders (VPEX):

- **Bridge Port Extenders (BPEs)**— Devices that do not fully process packets, nor make forwarding or filtering decisions. Instead, BPEs simply receive packets from extended ports and forward packets towards the upstream controlling bridge for L2/L3 processing.
- **Cascade port**—a port on a controlling bridge or BPE that connects to an upstream port.
- **Controlling Bridge (CB)**—In this context, a switch that provides full L2/L3 packet processing and filtering functionality for cascade ports and extended ports.

- **E-Channel-ID (E-CID)**—Identifies the BPE destination port (or port-list for multicast E-CID).
- **E-Tag**—A tag header immediately following the 802.1BR EtherType (0x893F) that contains the E-CID; this tag is only resident between bridge port extenders and controlling bridge devices.
- **Edge Control Protocol (ECP)**—Provides basic acknowledgement and re-transmission mechanism for reliable delivery; used for PE-CSP transmission.
- **Extended bridge**—A controlling bridge combined with one or more BPEs.
- **Extended ports**—Ports from BPEs connected to the rest of the network.
- **Port Extender Control and Status Protocol (PE-CSP)**—Simple command/response protocol that allows a controlling bridge to discover, program, and obtain status from a bridge port extender.
- **Upstream port**—Port on a BPE that connects to a cascade port.
- **Virtual Port Extender (VPEX)**—A switching architecture based on the *IEEE 802.1BR* specification, comprising a CB, and one or more BPEs.

Chassis Analogy

VPEX, with its controlling bridge (CB) and bridge port extenders (BPEs), is analogous to a chassis with slots.

When you enter VPEX mode (using `enable` **vpex**), the CLI prompt changes to show that the CB is in VPEX mode and is slot 1 in this context:

```
Slot-1 VPEX switch model #
```

In VPEX mode, you refer to ports in the familiar slot:port notation in applicable commands, where the CB is always slot 1, and the BPEs are slots 100–162 (see Figure 70).

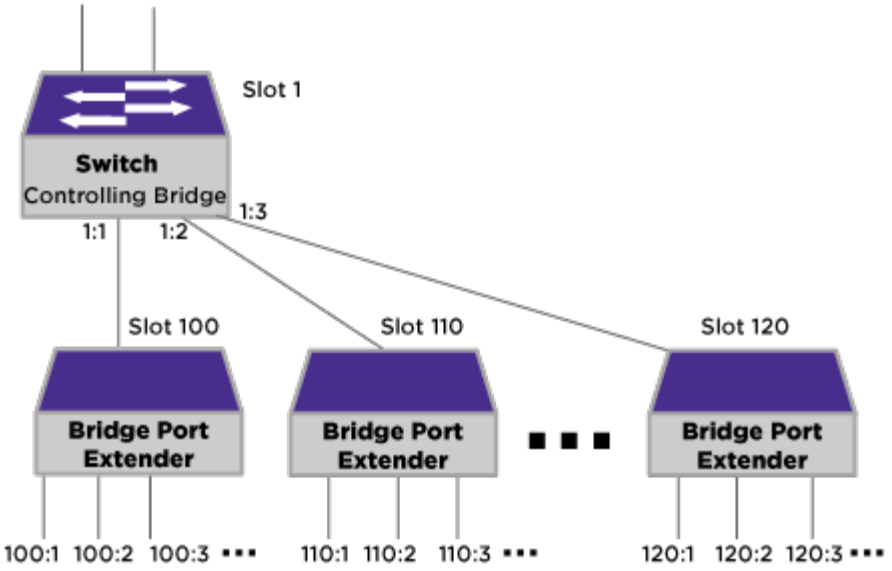


Figure 70: Extended Bridge — Chassis Analogy

The Value of Bridge Port Extenders

Since the bridge port extenders (BPEs) are managed like slots in a chassis under a single management domain, multiple layers of a traditional network can be reduced from a configuration and management perspective, greatly simplifying the network operation.

As opposed to expanding the network using additional switches, BPEs provide the following benefits:

- No console. BPEs are configured and managed through the controlling bridge (CB) user interface, so there is no need to connect a terminal console connection to each BPE.
- No out-of-band management.
- Single point of license management.
- Single point for configuring/debugging/diagnostics.
- Single node when managed by management software products.

How Do Bridge Port Extenders Interact with the Controlling Bridge?

Bridge port extenders (BPEs) are initially discovered using a type-length-value (TLV) extension to the Link Layer Discovery Protocol (LLDP) along with certain LLDP-MED TLVs that specify serial number, model name, hardware version, and firmware version. After initial discovery, the controlling bridge (CB) uses the Port Extender Control

and Status Protocol (PE-CSP) to control and manage the BPE. PE-CSP is a simple command/response protocol (see Port Extension Control and Status Protocol on page 739) that runs on top of the Edge Control Protocol (ECP), which provides reliable delivery (see Edge Control Protocol on page 738).

In the data-plane, the BPE receives Ethernet packets on its extended ports, inserts an E-TAG with an E-CID identifying the source port, and forwards it on its upstream ports towards the CB. When the CB receives a packet on a cascaded port, it uses the E-CID within the E-TAG to identify the source extended port and performs the L2/L3/policy forwarding decisions for the packet. Based on these forwarding decisions, the CB inserts an E-TAG to identify the destination and forwards the packet out the corresponding cascaded port. If the forwarding decision is to forward to another extended port, the CB inserts an E-Tag. The BPE receives the packet on its upstream port, processes the associated E-TAG, and forwards the packet out the corresponding egress extended port(s).

When the CB receives a packet with an unknown destination (unknown unicast, unknown multicast or broadcast), it inserts a special E-TAG assigned to the VLAN for flooding and sends out only one copy of the packet on the cascade port irrespective of the number of extended ports that need to transmit the packet. Each BPE performs the appropriate replication. If some of the recipients are members of extended port LAG, a BPE does not replicate a copy of the flood packet to them. Instead, the CB has to send an additional copy using the E-TAG assigned to one of the member ports of the extended port LAG. The CB selects one member of the extended port LAG to receive the packet based on hashing of packet fields.

Edge Control Protocol

The Edge Control Protocol (ECP) provides reliable transport for the Port Extender Control and Status Protocol (PE-CSP) protocol. It includes a basic acknowledgement and retransmission facility. The Edge Control Protocol Data Unit (ECPDU) structure (Figure 71) is slightly modified from the IEEE 802.1Q-2014. Preceding the ECPDU structure is the Destination and Source Address.

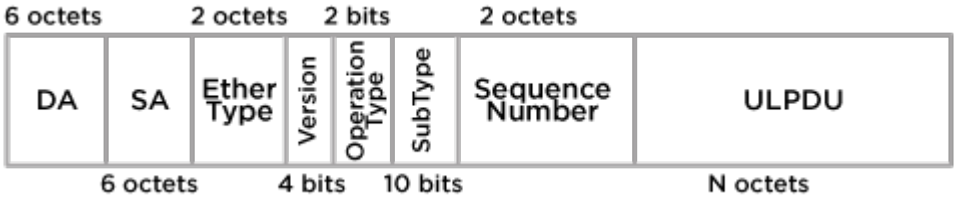


Figure 71: ECPDU Structure

- DA (Destination MAC address)—The destination MAC address of the Ethernet frame that contains an ECPDU is specified by the Upper Layer Protocol (ULP). When ECP is carrying PE CSP, the destination address used is the Bridge Group Address, nearest non-TPMR Bridge group address (01-80-c2-00-00-03) as listed in Table-8-1 of the *802.1Q-2014*. The DA is 6 octets long.

The BPE device accepts either the nearest non-TPMR bridge group address or the individual MAC address sent in its LLDP PE TLV.

- **SA (Source MAC address)**—The source MAC address of the bridge. The SA is six octets long.
- **EtherType**—ECP Ethertype (0x8940). The ECP EtherType is two octets long.
- **Version**—The ECP protocol version. The version is 4 bits long, and the value is 1.
- **Operation Type**—Defines the operation type. It is 2 bits long. The values are:
  - ECP Request (0x0)
  - ECP acknowledgment (0x1)
- **SubType**—Defines the ULP type included in the PDU. The value of Subtype is 0x2 when BPE CSP is the ULP. See table 43-1 of *IEEE-802.1Q-2014*. The SubType field is a 10 bits long.
- **Sequence Number**—The Sequence Number field identifies the sequential order of the PDU with respect to other ECPDUs. The first sequence number can start anywhere, but each subsequent sequence number is incremented by one. The Sequence Number is 2 octets long.
- **ULPDU**—The ULPDU contains the Upper Layer Protocol Data Unit if the operation type value is “ECP Request”. The ULPDU field is absent if the operation type value is “ECP Acknowledgment”. The length of this field is upper layer dependent. The ULPDU in this case is PE-CSP.

Port Extension Control and Status Protocol

The Port Extension Control and Status Protocol (PE-CSP) is a simple request/response protocol that runs over ECP and is initiated upon detection of a bridge port extender using the LLDP protocol. Each controlling bridge (CB) has a single PE-CSP protocol instance on each cascade port. In the case of a cascade port that is a LAG, the PE-CSP instance runs on the LAG and transmits on the current master port in the LAG. There is a separate PE-CSP session per LAG member port, but configuration commands are sent over a chosen single port in the LAG.

Each PE-CSP PDU is made of a command TLV and zero or more additional TLVs.

Figure 72 shows a typical PE-CSP exchange between a CB and a bridge port extender (BPE).

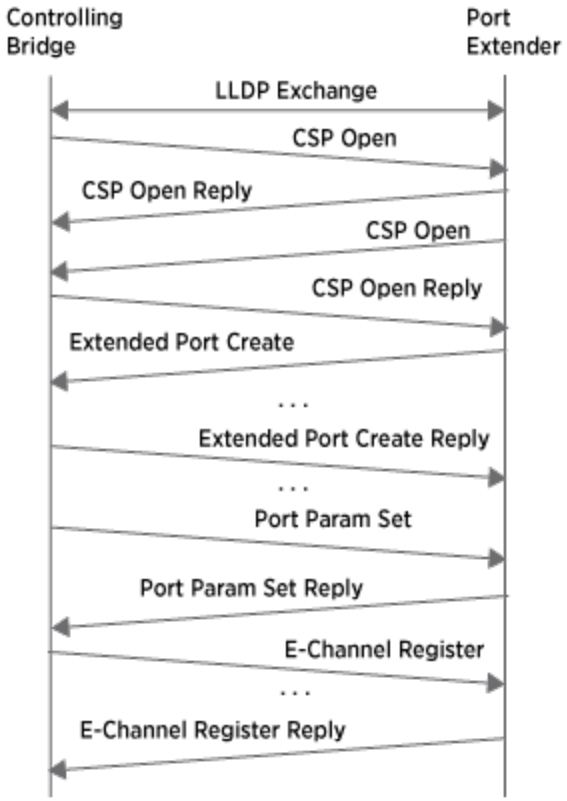


Figure 72: Example PE-CSP Protocol Exchange

Redundant Controlling Bridges

For added resiliency and redundancy, the VPEX architecture allows you to form an extended bridge using multiple controlling bridges (CBs).

You can use MLAGs to form an extended bridge with two CBs, also serving as MLAG peers, connected to dual-homed BPE devices using MLAG cascaded ports (see Figure 73 on page 741).

Note
When using MLAG, you must run LACP on the CB MLAG cascade ports.

Note
With an MLAG/Extended Edge Switching topology, enabling policy and using policy features does not work with the ONEPolicy default resource profile. Use the profile-modifier feature to free up ACL resources from any specific policy profile (see Platform Rule Allocation on page 1349).

Note
Extended Edge Switching Auto-configuration (see Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760) uses values within the range of 5100 to 5164 to configure an MLAG port ID. It is recommended that you do not use values within this range when configuring MLAG port IDs manually, since the MLAG port might be unconfigured by auto-configuration when auto-configuration finds the MLAG port ID has already been used.

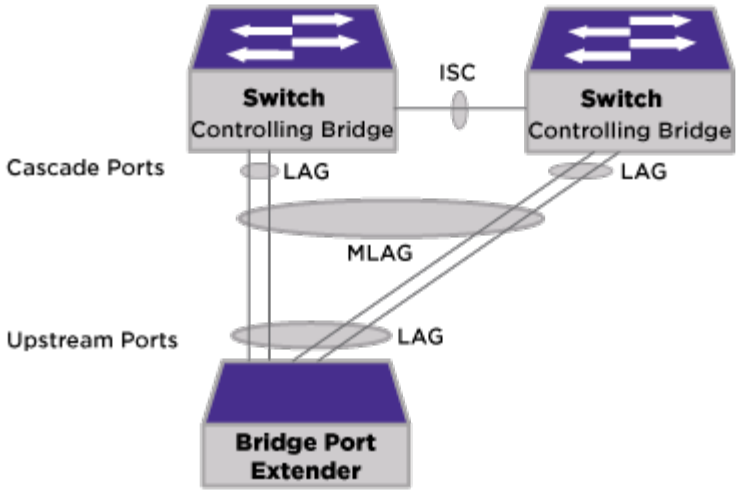


Figure 73: Redundant Controlling Bridge Architecture with MLAG over Cascaded Ports

With redundant CBs attached to each BPE, the associated extended port configuration must be identical on each controlling bridge. To reduce the configuration complexity and to minimize the risk of inconsistency, you can use orchestration mode so that any configuration commands are now checkpointed to the MLAG peer switch. Before entering orchestration mode, ensure that any configuration parameters (connecting ports for the BPEs, VLAN names, numbers etc.) are the same for both CBs.

After entering this orchestration mode, like the existing virtual-router mode, the configuration prompt changes indicating that commands issued are within this context:

```
(orchestration bottom) switch model #
```

When using automation (see Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760 and Configuring Extended Edge

Switching Topology with Full Automation on page 757) to set up redundant CB MLAGs, the following occurs:

- The same slot number is assigned to the same BPE on both CBs.
- MLAG port is configured for first tier BPEs. The MLAG port identifier is set to 5,000 plus the BPE's slot number.

To enter orchestration mode, use the following command:

```
start orchestration mlag peer_name
```

To exit orchestration mode, use the following command:

```
stop {orchestration}
```

Extended Edge Switching Rings

Extended Edge Switching rings allow two Extended Edge Switching (VPEX) cascades to be joined together to form a control plane ring. If a link breaks or a bridge port extender (BPE) otherwise leaves, the remaining BPEs reform two data plane cascades, thus keeping both control and data plane connectivity to the controlling bridge (CB) alive. This provides a redundant connection from any BPE in the ring to the CB. This is especially useful in a wiring closet application where the BPEs are located in the closet, and the CB is located more centrally, and there are only two links wired from the CB to the closet. Each cascade is formed from the control plane perspective only; the data plane acts as if there were two cascades consisting of BPEs that are each present in only one cascade. You can only achieve full redundancy with redundant CBs (see Redundant Controlling Bridges on page 740).

Rings are not explicitly configured, but rather are detected and created automatically. A ring contains the BPEs that were used in its formation. The life of a ring is ended by either a configuration command that changes one or both cascades, or a reboot of the CBs.

To support a wiring closet with only two links, each BPE that directly connects to the CB connects with only one link in its . To support this capability without forcing you to reserve and not use a port on one of the CBs, ExtremeXOS supports “one-arm” MLAGs where one CB has a connection reserved, and the other does not (see One-Arm MLAGs on page 746). This allows MLAG operation as if one link were down, meaning that both CBs are able to switch to the MLAG. ExtremeXOS also supports regular cross-connected MLAG at the top of the ring.

Multiple rings are supported. However, each ring must be completely separate from all others. For example, a simple ring could be formed from a single CB ports 1:1 and 1:2, while a separate ring could be formed from ports 1:3 and 1:4. Multiple rings cannot otherwise intersect, except at the CB itself.

Limitations

- A maximum of 8 BPEs per ring is supported.
- A ring can be formed from exactly one or two cascades. A cascade (or any part thereof) can be a part of at most one ring.

- Three-way MLAG is not supported.

For information about setting up a ring, see Configuring Extended Edge Switching Ring Topology on page 762.

Forming a Ring

To form a ring, you configure one or two cascades and allow them to form. After that, you connect the end bridget port extenders (BPEs) of these cascades to each other, or the end of the single unconfigured cascade to a port on the controlling bridge (CB). After establishing the control plane, the formed cascades are as shown in Figure 74.

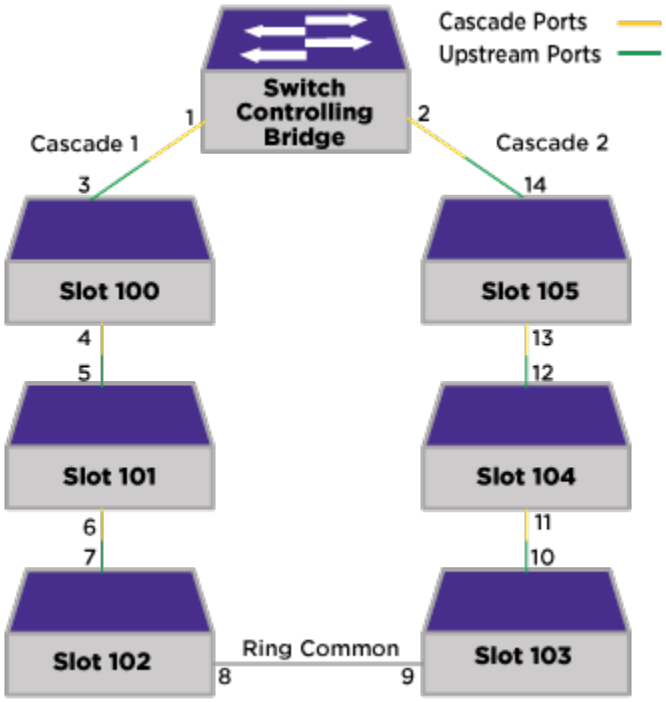


Figure 74: Simple Extended Edge Switching Ring

In Figure 74, the “ring common” link between slot 102 port 8 and slot 103 port 9 forms the ring between the two cascades. In a complete ring, the ring common link does not carry data plane traffic. When the cascades were formed, both BPEs have active upstream ports. Since slot 102 port 8 and slot 103 port 9 are up, both begin sending *LLDP (Link Layer Discovery Protocol)* packets. Using LLDP, the connection between cascade slot 100 to 102 and cascade slot 105 to 103 is discovered, which initiates the ring detection and configuration process.

The completed control plane on a ring establishes two counter-directional cascades with backup cascade and upstream ports that are dynamically configured (Figure 75). While the CB retains knowledge of a ring even when the ring is severed, the BPEs do not. A BPE is only in ring mode when the ring is being brought up or is complete. When not in ring mode, a BPE does not have a backup upstream port or any forwarding information associated to any backup direction.

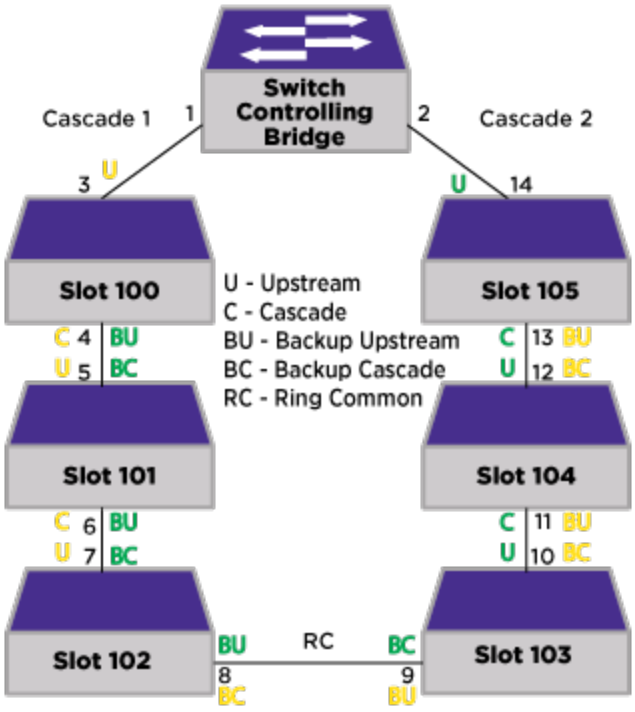


Figure 75: Counter-directional Cascades on Completed Ring

Data Plane Establishment and Operation

The data plane is established initially as two separate and distinct cascades as shown in Figure 74 on page 743. After the ring control plane is established, the data plane remains the same, except that the cascades are extended in both directions around the ring to facilitate failover if a link break occurs.

Link Failure Handling

When a link in the ring goes down, the ring splits into two separate cascades. This occurs in the following sequence of events:

1. Link Failure Detection and the Ring Link Fail Notification (RLFN) Message
2. Ring Disable Message
3. Traffic Re-routing and Cascade Information Clearing
4. Failed Link Restoration

When the connection between two BPEs in a ring is severed, the BPE whose upstream port was at the point of disconnection sends a notification of the failure to its active cascade port. Simultaneously, that BPE will "fail over" to its backup upstream port, changing the upstream direction of the data plane. The notification is called a Ring Link Failure Notification (RLFN). When the downstream BPE receives it, it forwards the RLFN to its active cascade port. If the BPE received the RLFN over its active upstream port, it too fails over as above. The end result is one or two data plane cascades terminating at the broken connection with no remaining backup path, which is re-established after the next ring detection. The new backup upstream direction is established on all nodes. Some BPEs will have different upstream and backup

upstream ports than those that were originally established. The cascades that were established when the ring was severed remain in place.

Uneven Cascade Lengths and Ring Re-balancing

When a ring is formed from two unequal length cascades, or re-formed due to a ring link break and restoration, there is a desire to equalize the cascades from a data plane perspective. This reduces traffic flow over one cascade at the expense of the other. This becomes especially important when a ring has been severed resulting in two unequal length data planes, and then rejoins. Although not described in Forming a Ring on page 743, rebalancing always take place after control plane ring formation. By always re-balancing, regardless of how the two cascades are configured, the result is two approximately equal length data plane cascades. By default, ring re-balancing is disabled; however, you can globally enable all ring re-balancing.

Dual Controlling Bridges in Ring Topologies (MLAGs)

In an Extended Edge Switching ring topology with dual (redundant) controlling bridges (CBs), s are used to synchronize and coordinate the ring state between the two CBs. For more information about redundant CBs, see Dual Controlling Bridges in Ring Topologies (MLAGs) on page 745.

Two MLAG topology types are supported:

- **Cross-connect MLAG**—requires 48-port bridge port extenders (BPEs) with four trusted ports (for example, V400-48p-10GE4). See Cross-Connect MLAGs on page 745.
- **One-armed MLAG**—used for wiring closet with existing cabling limitations or smaller BPEs. See One-Arm MLAGs on page 746.

Cross-Connect MLAGs

For dual (redundant) controlling bridges (CBs) on Extended Edge Switching rings, you can use cross-connect s. Cross-connect MLAGS require 48-port bridge port extenders (BPEs) with four trusted ports (for example, V400-48p-10GE4). Figure 76 shows a ring formed using CBs connected to a BPE ring using two cross-connect MLAGs.

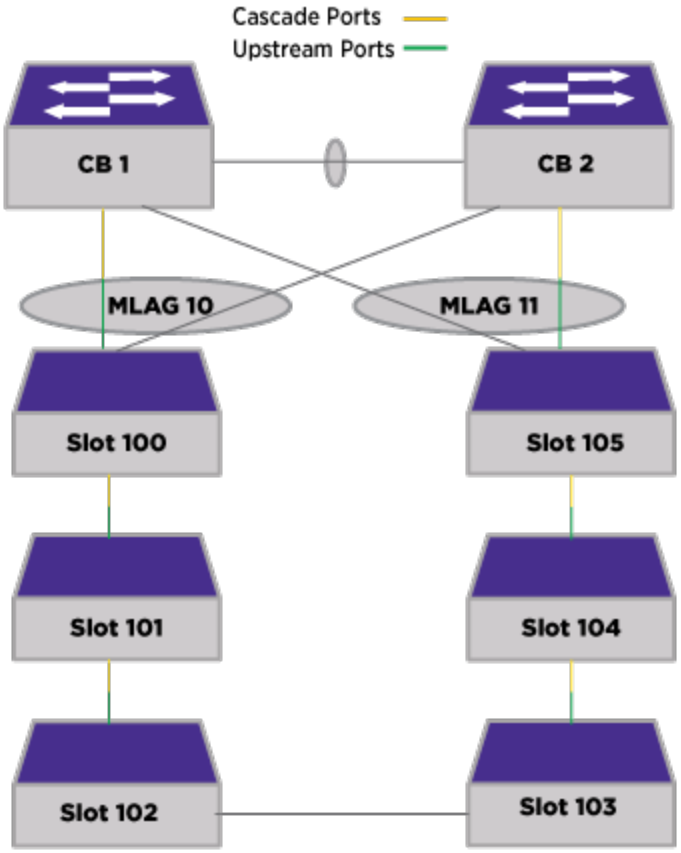


Figure 76: Cross-Connect MLAG

One-Arm MLAGs

In some wiring closet applications, bridge port extenders (BPEs) are located in the closet, and the controlling bridge (CB) is located more centrally, and there are only two links wired from the CB to the closet. An important capability of Extended Edge Switching rings is to provide redundancy among BPEs that have only two trusted ports. In this configuration the CB neighbor has only one link to one of the cascades of the ring. To overcome this limitation, you can use a one-arm to achieve CB redundancy. Figure 77 shows a ring formed using CBs connected to a BPE ring using two one-armed MLAGs. One-armed MLAG uses virtual MLAG ports where a physical port connection is not present. The virtual port is always down, so MLAG blocking rules are not applied.

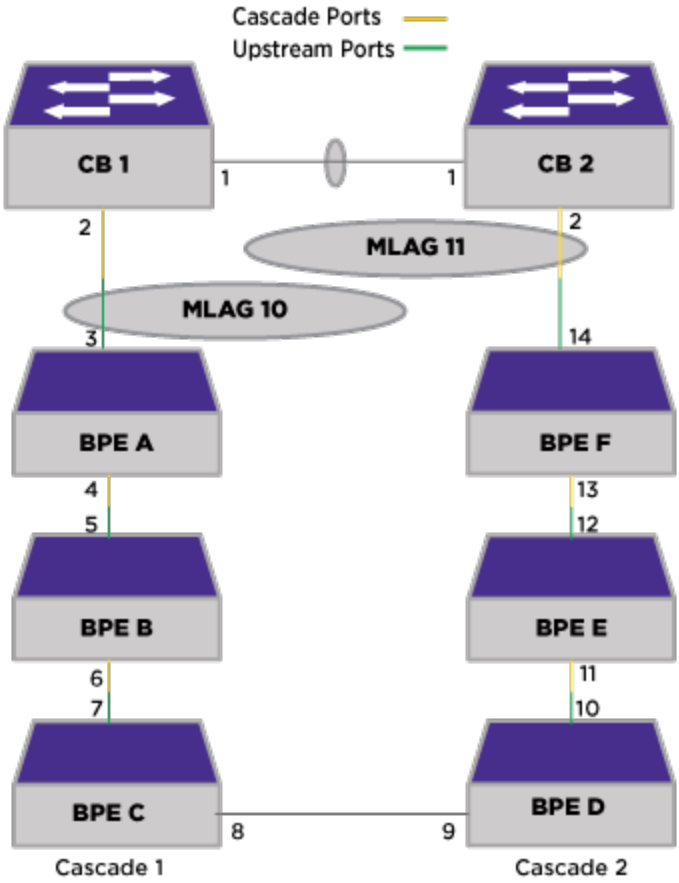


Figure 77: One-Arm MLAG

CB-1 and CB-2 are both aware of both MLAGs.

On CB-1, Cascade 1 is configured with port 2 as the native cascade port, and similarly, on CB-2, Cascade 2 is configured with port 2 as the native cascade port.

On CB-1, Cascade 2 is configured to use MLAG 11. On CB-2, Cascade 1 is configured to use MLAG 10. Such configurations are retained by the local CB even if the specified MLAG does not exist when the configuration is first processed.

When the first MLAG comes up, CB-1 and CB-2 elect a master that is the controller of all CSP sessions with BPEs that reside at any tier other than tier 1 (the BPEs directly connected to the CB). The same mastership applies to both MLAGs. Because the MLAGs are one-armed, the master sends messages to such BPEs through the CB that is directly connected to tier 1 of the related cascade. You can determine which CB is the master using the `show` **`vpex ports`** *`{ports_list}`* command.

Each CB establishes and controls the CSP session with the tier 1 BPE that is locally connected. The master CB establishes and controls all CSP sessions with BPE that are not at tier 1.

Optimized IP Multicast Replication

With optimized IP multicast replication, a controlling bridge (CB) only sends one copy of the IP multicast traffic per receiver VLAN for a particular IP multicast group. For example, if group 225.0.0.1 has five receivers in VLAN v1 and four receivers in VLAN v2, the CB sends one copy to VLAN v1 and one copy to VLAN v2 and the bridge port extender (BPE) does the replication.

If the receivers of VLAN v1 (or VLAN v2) are spread across different native cascade ports, separate copies might be sent for VLAN v1 on each native cascade port depending on the best path to reach a receiver. However, when the receiver is an extended port LAG, the CB has to do the replication. Because extended port LAG members could belong to different BPEs, and because BPE cannot do the pruning, the CB has to send copies to the individual LAG member. For instance, if VLAN v1 membership includes a LAG comprising ports 101:1, 103:1, and another LAG comprising ports 102:1, 102:2, two copies are sent, one for the 101:1 LAG member and another one for the 102:1 LAG member.

Because replication and VLAN flooding both use multicast ECIDs, using optimized IP multicast reduces the number of VLANs with extended ports to 3,000.

Supported Bridge Port Extender Models

The following V400 bridge port extender (BPE) models are available:

Table 41: V400 BPE Models

Model Numbers	Description
V400-24t-10GE2	V400 Series 24 10/100/1000BASE-T, 2 1000/10GBaseX unpopulated SFP+ ports, fixed power supply and fan.
V400-24p-10GE2	V400 Series 24 10/100/1000BASE-T PoE+, 2 1000/10GBaseX unpopulated SFP+ ports, fixed power supply and fans, and optional redundant power supply.
V400-48t-10GE4	V400 Series 48 10/100/1000BASE-T, 4 1000/10GBaseX unpopulated SFP+ ports, fixed power supply and fan.
V400-48p-10GE4	V400 Series 48 10/100/1000BASE-T PoE+, 4 1000/10GBaseX unpopulated SFP+ ports, fixed power supply, fans, and optional redundant power supply.

Important
On V400 BPEs, only SFP+ ports can be used for upstream/cascade ports.

The following V300 BPE models are available:

Table 42: V300 BPE Models

Model Numbers	Description
V300-8P-2T-W	V300 Series 8 port 10/100/1000BASE-T POE+ ports half/full duplex, 2x1000 BASE-T ports, PoE powered 802.3bt Type 4, fanless. **Note:** The V300-8P-2T-W model can only be directly attached to a controlling bridge. It cannot be cascaded.
V300-8P-2X	V300 Series 8 port 10/100/1000BASE-T POE+ ports half/full duplex, 2 × SFP+ ports, fanless.
V300-8T-2X	V300 Series 8 port 10/100/1000BASE-T ports half/full duplex, 2 × SFP+ ports, fanless.
V300HT-8P-2X	V300 Series 8 port 10/100/1000BASE-T POE+ ports half/full duplex, 2 × SFP+ ports, fanless, high-temperature model.
V300HT-8T-2X	V300 Series 8 port 10/100/1000BASE-T ports half/full duplex, 2 × SFP+ ports, fanless, high-temperature model.

Important
On V400 BPEs, only SFP+ ports can be used for upstream/cascade ports.

Important
On V300 BPEs, only ports 9 and 10 can be used as upstream ports.

For information about which optics are supported with the Virtual Port Extenders, see the *Extreme Hardware/Software Compatibility and Recommendation Matrices*.

Supported Platforms

MSDP (Multicast Source Discovery Protocol) is supported on all platforms running a minimum software version of ExtremeXOS 12.0 with the Premier or Extreme Platform ONE license.

Our implementation of MSDP is compliant with RFC 3618 and RFC 3446, and compatible with other devices that are compliant with these standards.

Extended Edge Switching on Stacking Overview

With Extended Edge Switching on Stacking, Extended Edge uses the stack, rather than MLAG, to provide redundancy to a BPE. This is done by having certain components of EXOS synchronize their primary and backup nodes with respect to the Extended Edge Switching state. The primary node performs all Extended Edge Switching controlling bridge functions while the backup is kept in sync with the primary. This simplifies configuration, as all stack switches are managed with a single configuration file and control point that resides on the elected primary switch. The backup switch can

become the primary switch upon failure of the primary switch if the failure is caused by a crash, a power-off, or a management action.

Use of MLAG in Extended Edge Switching on Stacking

When Stacking provides the controlling bridge, MLAGs will not service individual BPEs. All BPE LAGs that connect to a controlling bridge provided by a stack terminate in a single LAG at the controlling bridge. The LAG at the controlling bridge can consist of member ports provided by different nodes in the stack. Preventing Extended Edge Switching MLAG in Stacking is enforced by the CLI, and ZTPStack will not attempt to activate MLAG on the stack.

MLAG can be used to provide redundancy using two stacks. An MLAG is defined for a user device using stack node front panel ports that are not used to provide the cascade port function.

Note
BPE ports cannot be used to form MLAGs.

Extended Edge Switching and Stack Modes Combined

Either Extended Edge Switching mode or stacking mode can be used independently. A switch reboot is required to enter the combined mode and enable Extended Edge when already in stacking mode. A reboot is also required to enable stacking mode when in Extended Edge Switching mode (assuming that Extended Edge Switching MLAG is not in use). Both can be enabled simultaneously with a single reboot.

Extended Edge Switching Autoconfiguration and ZTPStack

ZTPStack enables both Extended Edge Switching and stacking when bringing up a stack that has attached BPEs. Once in stacking mode and Extended Edge Switching mode simultaneously, ZTPStack (on stacks that have a default EXOS configuration) enables Extended Edge autoconfiguration.

Supported Topologies

BPEs at the “first tier” (directly connected to the stack) can connect any or all of their intended upstream ports to any switch in the stack.

A cascaded BPE (tier 2 or higher) can connect its upstream ports only to the same BPE, meaning only point-to-point upstream topologies are supported below tier 1. Cascade ports that are configured on LAGs impose the restriction that all of the LAG members must be provided by the same BPE, or must all be provided by the stack directly.

Two BPE cascades can be connected together at their ends forming an Extended Edge Switching Ring.

Redudancy and Hitless Failover

Extended Edge Switching on Stacking uses the stack redundancy feature. All of the data and states associated with the Extended Edge Switching feature are continuously checkpointed to the backup node. The backup node is prepared to take over the controlling bridge function for the stack if there is a primary node failure.

Limitations

All nodes in a stack running Extended Edge Switching must be controlling bridge capable. Any attempt to add a non-capable slot to a stack running Extended Edge will fail. Enabling Extended Edge Switching on a stack that contains a non-capable node will fail.

You must use the same class of controlling bridge for both the primary and backup. For example, if a 5520 series switch is the primary node, then the backup must also be a 5520 series switch.

The primary-capable node in a stack that has the least maximum supported number of BPEs determines the maximum for the stack. When using Extended Edge with Stacking, BPEs cannot be attached to configured MLAG ports.

Supported Platforms

The following devices are supported as controlling bridges (CBs):

ExtremeSwitching 5420 and 5520, and Extreme 7520-48Y series switches.

The V300 and V400 BPEs can be attached to a stack that is configured as a controlling bridge.

Note
The V300-8P-2T-W BPE requires PoE.

Supported Commands

Existing Extended Edge display commands show the same result on the primary and backup nodes.

The **`enable/disable vpex`** and **`enable/disable stacking`** commands are aware of the capability of each node in the stack. The **`enable vpex`** command will fail if any node in the stack is not Extended Edge-capable.

The **`show slot`** command shows the Failed state for the attempted addition of a new stack node that does not support the controlling bridge function when the Extended Edge Switching feature is enabled. A message is logged at the time this failure occurs. The **`show slot`** command also shows the presence of any BPEs that have occupied slots.

Use the **show vpex stacking** command to show Extended Edge Switching information for all nodes in the stack topology.

ZTPStack Behavior

ZTPStack runs on all Extended Edge switches that have not yet been configured. ZTPStack is a function that either sets up SummitStack, or detects BPEs and sets up MLAG and enables Extended Edge Switching and Exteded Edge autoconfiguration.

With Extended Edge Switching on stacking, ZTPStack prefers stacking operation to MLAG. Controlling bridge capable switches that are normally deployed in stacking environments will default to stacking-support enabled mode.

When stacking-support is enabled (by default on certain switches), ZTPStack will enable Extended Edge Switching upon detection of an attached BPE, but will not reboot the switch. Stacking automation (push button or easy-setup) completes this task.

Software Upgrade

The package containing EXOS and the supported BPE image can be loaded on a stack (the `.lst` file). The installation of the images is performed on all stack nodes in the active stack topology.

Bridge Port Extender General Limitations

- In the Extended Edge Switching architecture, Layer-2, Layer-3, multicast, and filtering operations take place on the controlling bridge. The controlling bridge switch and attached BPEs constitute a single, extended switch system. Therefore, the Extended Edge Switching system assumes the scale and limits from the specific controlling bridge model in use. For applicable limits, see the Limits tables in the *ExtremeXOS Release Notes*.
- Certain ExtremeXOS features are not supported or have limitations. For information, see ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders on page 753.
- A maximum of 48 attached BPEs is supported (20 BEPs for 5420 controlling bridge).
- A maximum of 4 levels (tiers) of cascaded BPEs is supported.
- A maximum of 4 upstream ports (in a LAG) per BPE is supported (with V400-48).
- Trusted ports on BPEs cannot be used as user data ports. No protocols or features should be applied/configured on this port.

ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders

Table 43 shows which ExtremeXOS features are supported, or are not supported, or that have limitations when used with V400 Virtual Port Extenders.

Table 43: ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders

ExtremeXOS Feature	Status
ACLs	Ingress (first and second stage) and egress ACLs are supported with the following limitations: • Ingress ACLs applied to a list of extended ports results in one hardware rule per extended port. This is the same behavior as regular ports when counters and/or meters are used. • Egress ACLs applied to an egress port do NOT match L2 flood traffic, but egress ACLs applied to VLANs do. • Ingress ACLs with Replace-ethernet-destination-address action, replace-ethernet-source-address, and replace-vlan-id actions are not supported when packets are egressing extended ports. • L3/L4 match criteria are not available in the same egress ACL key as an extended port match. • ACL byte counters include an extra 8 bytes per packet (Ethertype + Etag) for the added encapsulation. • Egress ACLs applied to extended port LAGs are not supported. Use egress ACLs bound to VLANs/any instead. **Note:** When installing ACL rules bound to a set of ports, rules are replicated for each port if there are ACL counters and counter compression is not enabled or if the ports are extended ports. **Note:** You can consume an excessive number of ACLs if you bind them to each individual (extended) port in cases where you have many extended ports. To avoid this, consider policy (role-based assignments per user) for access-control (see Policy Roles on page 1340), ACLs bound to VLANs, wildcarded applications (see Configuring the ACL Rule on the Interface on page 372), or two-stage ACLs (see Two-Stage ACL on page 342).
AVB	Not supported.
BFD	Supported, except on extended ports.
CFM	Supported, except on extended ports.
Diffserv examination	Supported.
Disable slot	Not supported. **Note:** You can take a BPE slot offline by disabling the corresponding cascade port(s).
Dot1p examination	Disabled by default on extended ports; enabled by default on controlling bridge cascade ports. Use cascade port Dot1p examination for extended port traffic to conserve ACL hardware resources.

Table 43: ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders (continued)

ExtremeXOS Feature	Status
EAPS	Supported, except that EAPS ring ports cannot be extended ports.
EDP/LLDP/CDP	Supported. **Note:** BPEs will not appear as EDP neighbors.
EEE	Supported.
ELSM	Not supported.
ERPS	Supported, except that ERPS ring ports cannot be extended ports.
ESRP	Not supported.
Fabric Attach	ExtremeXOS Fabric Attach (FA) supports FA servers and FA Proxies to be Extended Edge Switching controlling bridge (CB) nodes connected with bridge port extenders (BPEs) in two different topologies: • Single-arm ring—see Single-ARM MLAG Ring Topology Configuration Examples. • Cross-connect MLAG—see Cross-Connect MLAG Topology Configuration Examples.
FDB	
FDB: Limit learning	Supported.
FDB: MAC lockdown	Supported.
FDB: MAC locking	Not supported on Extended Ports.
FDB: Static and blackhole entries	Supported, except that multi-port unicast FDBs entries are not supported.
Flood rate-limit	Not supported on extended ports.
HcLAG	Not supported.
Identity Management	Identity Management should not be enabled on cascade ports.
L2 PBR	Not supported.
FDB: multi-port unicast entries	Not supported.
L3	
L3: PBR	Supported.
L3: Gateway routers	Gateway router connected to extended port LAGs is not supported.
L3: IP unicast forwarding	Maximum gateways (using `configure iproute sharing max-gateways` *`max_gateways`*) must be greater than or equal to the number of active extended LAG members.
L3: Routing protocols	Supported on CB; not supported on extended ports.

Table 43: ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders (continued)

ExtremeXOS Feature	Status
L3 Tunnels	Supported on CB ports. Not supported on extended ports.
LAG: LACP	Supported. The default and only supported algorithm on extended port LAGs is "custom."
V400 Virtual Port Extender LEDs	Supported. For information about the location of BPE LEDs, see the *V400 Virtual Port Extender Quick Reference Guide*.
Mirroring	Mirroring extended ports is supported. Mirroring to extended ports is supported on all Universal platforms. Ingress mirroring of an extended port will include an 802.1BR E-tag. Note: Workaround: You can avoid this 802.1BR E-tag by doing the following: 1. Run the command: `enable mirroring to port-list 1:1 loopback 1:2`. Where in this example, port 1:2 is not connected to anything. 2. Run the command: `configure mirroring add port 104:1`. Where in this example, port 104:1 is the extended port. Mirroring to BPE cascade ports is not supported. Limitations: Only known traffic to extended ports can be egress mirrored.
MLAG	Supported on cascade ports; not supported on extended ports. Not supported on cascade ports when in stacking mode.
MPLS/VPLS	Not supported.
Multicast	
Multicast: IGMP	Supported.
Multicast: MVR	Supported.
Multicast: PIM	Supported.
Multicast: PIM Snooping	Supported.
MVRP	Supported.

Table 43: ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders (continued)

ExtremeXOS Feature	Status
Netlogin	Dot1X and MAC Auth supported, In VPEX-MLAG implementations, Dot1X and MAC Auth are supported only in Policy mode. Not supported: • Multi-supplicant (MAC-based VLANs) without ONEPolicy. • Port restart option. • Trusted ports. WebAuth is not supported with MLAG.
Node Alias	Supported.
OnePolicy	Basic policy supported: • Policy scale (number of users, roles, rules) same as CB. • LAG rules not supported on extended ports with tci-overwrite. • Per-port, policy-driven, ingress rate limiting is not supported on extended ports. • 5420 and 5520 series switches as CBs do not support tci-overwrite.
PoE	Supported on select BPE models. See Supported Bridge Port Extender Models on page 748. Operator limit configuration is not supported on cascade ports.
Ports	
Ports: Flooding	Disabling flooding is supported.
Ports: Jumbo frame	Enabling/disabling jumbo support on extended ports is supported; cascade-ports default to jumbo enabled. The max-frame-size on V400 is enforced only on ingress, and not on egress.
Ports: Learning	Disabling/enabling learning supported.
PVLAN	Not supported.
QoS	Supported: • 8 queues per extended port. • Peak-rate supported per qosprofile per extended port. • CBS is 32 Kbytes. • Scheduling algorithms WDRR and strict priority. • Egress per-port rate limit per extended port. Not supported: • Committed-rate per qosprofile per extended port. **Note:** Dot1p examination is enabled by default on cascade ports, but disabled by default on extended ports. **Note:** When using hyrbid QoS scheduling, strict-priority queues must be the highest number queues.

Table 43: ExtremeXOS Feature Compatibility with V400 Virtual Port Extenders (continued)

ExtremeXOS Feature	Status
Restart port	Not supported on extended ports. Use `disable/enable port` commands.
sFlow	Supported: sFlow hardware sampling on Extended Edge Switching extended ports is provided by the underlying Extended Edge Switching cascade port hardware. As such, the configured sample rate is programmed on the cascade port and is provided in aggregate to all subtended sFlow-enabled Extended Edge Switching extended ports.
Software-redundant ports	Supported.
STP	Supported, primarily for edge loop detection (edge guard); supported with MLAG redundancy for RSTP and MSTP protocol modes.
VLAN	
VLAN:Membership	Supported; users specify VLAN membership using <slot>:<port> notation; protocol VLANs not supported.
VLAN: statistics	Supported on extended ports. **Note:** VLAN flood traffic is not counted in the transmit counters; displayed counts include an additional 8 bytes for the Ethertype + E-tag encapsulation.
VMAN/CEP	Not supported.
VMT/XNV	Not supported.
VRRP	Supported.
VXLAN	Supported for tagged and untagged tenant VLANs on extended ports, except for 5420 and 5520 series switches. Extended port LAG in VXLAN is not supported. Extended ports cannot be in VXLAN network VLANs. Extended ports cannot participate in RIOT configurations. Ring is not supported for VXLAN extended ports. Optimized IP Multicast Replication is not supported for VXLAN extended ports.

Using VPEX Bridge Port Extenders

Configuring Extended Edge Switching Topology with Full Automation

You can automatically configure an Extended Edge Switching topology fully or in part. This capability allows for the more rapid and accurate setup of an Extended Edge Switching architecture.

However, the disadvantage of full automation is that you must start with an unconfigured controlling bridge (CB) (out of the shipping box) or unconfigure the CB switch. To avoid unconfiguring the CB, but to still avoid manually configuring the entire Extended Edge Switching topology, consider using partial automation (see Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760).

Note
Do not attach bridge port extenders (BPEs) to a partitioned port. Unconfiguring the CB removes the port partition configuration, and this causes full automation to fail.

Full automation for Extended Edge Switching performs the following tasks:

1. Determines if the switch is capable of being a CB.
2. Detects if any BPEs are attached.
3. Enables VPEX mode on the CB.
4. If you are setting up redundant CBs (see Redundant Controlling Bridges on page 740):
   - Detects the other CB.
   - Creates a VLAN on only the port(s) between the CBs.
   - Adds link local IP address to the VLAN.
   - Creates a LAG for the port(s) between the MLAG peers.
   - When the CB detects the IP address of the other CB VLAN, creates the MLAG peer.
5. Enables Extended Edge Switching auto-configuration (see Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760, which performs the following tasks:
   a. Assigns the next available slot number to each BPE.
   b. Creates LAGs/MLAGs on cascade ports.
   c. Adds ports to existing cascade port LAGs.
6. Defaults to stacking-support enabled mode with Extended Edge Switching on stacking.

Note
ZTPStack will enable Extended Edge Switching upon detection of an attached BPE, but will not reboot the switch. Stacking automation (push button or easy-setup) completes this task.

Note
For full automation, the entire process can take about 6–7 minutes to complete depending upon the configuration.

To start full automation for Extended Edge Switching:

1. (Optional) If you are setting up redundant CBs, connect the two CBs to each other.
2. Attach BPE(s).

3. Ensure that full automation is enabled. Full automation is enabled by default. Run `show vpex`:

```
# show vpex
Virtual Port Extender: Disabled
Auto-Configuration:    Enabled
Ring rebalancing:      Auto
Zero Touch Provisioning: Nothing Provisioned - Existing configuration present

Cascade
Port/MLAG Id    Slot
====================
```

   You should see `Zero Touch Provisioning: Nothing Provisioned - Existing configuration present`. If you see `Nothing Provisioned - Disabled`, run `restart process ztpstack`.
4. Ensure that none of the following files are present on the CB(s):
   - `default.xsf`
   - `default.py`
   - `autoexec.xsf`
   - `autoexec.py`

   For information about locating and deleting files on the switch, see Using the ExtremeXOS File System on page 135.
5. Ensure that the CB(s) do not have a configuration. Run `unconfigure switch {`**`all`** `|` **`erase`** `[`**`all`** `|` **`nvram`**`]}`. Use either:
   - `unconfigure switch`—removes configuration, but leaves other settings intact.
   - `unconfigure switch all`—removes configuration and resets all settings to factory defaults (recommended).
6. Reboot the CB(s).
7. The configuration applied by automation is not automatically saved. Save the configuration:
   - Manually: `save configuration {`**`primary`** `|` **`secondary`** `|` *`existing-config`* `|` *`new-config`*`}`
   - Automatically: `save configuration` **`automatic`** `{`**`every`** *`minutes`* `{`**`primary`** `|` **`secondary`** `|` *`existing-config`* `|` *`new-config`*`}` `|` **`never`**`}`

For information about troubleshooting an Extended Edge Switching topology, see Troubleshooting an Extended Edge Switching Topology on page 764.

Note

To avoid having automation start if you do want it to, run `terminate` **`vpex ztp`** or perform some configuration prior to attaching BPEs or have a `default.xsf` (or other similar file) file in place.

To view automation status, use the command `show vpex`.

Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration)

The Extended Edge Switching auto-configuration feature allows the controlling bridge switch in an Extended Edge Switching topology to detect new bridge port extenders (BPEs) connected to ports not configured as cascade ports, and automatically configure cascade ports, extended slots, and LAGs/MLAGs on cascade ports. This capability allows for the more rapid and accurate setup of an Extended Edge Switching architecture, but still requires some manual configuration, in particular . To further reduce manual configuring, consider using full automation (see Configuring Extended Edge Switching Topology with Full Automation on page 757)

Note

Auto-configuration is disabled by default. If auto-configuration is enabled, Extended Edge Switching ports cannot be manually configured.

Cascade port connections to a BPE are automatically grouped into a LAG with the first detected connection set as the master logical port. Automatic LAG grouping occurs even if there is only a single connection to the BPE. With auto-configuration enabled, additional connections added after the initial Extended Edge Switching configuration is established are automatically added to the LAG. If you remove connection(s) and want the port(s) removed from the LAG, you must do this manually (see Adding and Deleting Ports in a Load-Sharing Group on page 282).

To use auto-configuration:

1. Attach BPEs as desired.
2. (Optional) Create the MLAG peer relationship between redundant CBs and ensure that both CBs are up (see Redundant Controlling Bridges on page 740). The rest of the MLAG setup for redundant CBs is handled automatically.
3. Enable VPEX mode:

   ```
   enable vpex
   ```
4. Reboot the CB(s) to allow the VPEX mode command to take effect.
5. Enable auto-configuration:

   ```
   enable vpex auto-configuration
   ```
6. Check that the intended cascade/ring configuration has been properly established:

   ```
   show vpex topology { port port_num} {summary | detail}
   ```
7. The configuration applied by automation is not automatically saved. Save the configuration:
   - Manually: `save configuration {primary | secondary | existing-config | new-config}`
   - Automatically: `save configuration automatic {every minutes {primary | secondary | existing-config | new-config} | never}`

The partial automation process detects the new BPEs connected to ports not configured as cascade ports, and automatically configures cascade ports, extended slots, and LAGs on cascade ports.

To show auto-configuration status, use the following command:

```
show vpex
```

To disable auto-configuration, use the following command:

```
disable vpex auto-configuration
```

For information about troubleshooting an Extended Edge Switching topology, see Troubleshooting an Extended Edge Switching Topology on page 764.

Manually Configuring Bridge Port Extenders

Tip

For an easier way to configure a Extended Edge Switching setup, use full automation (see Configuring Extended Edge Switching Topology with Full Automation on page 757) or partial automation (see Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760).

Use manual configuration when you want precise control over items such as slot number assignment.

To manually setup up Extended Edge Switching:

1. Physically set up the bridge port extender(s) (BPEs). See the applicable *Virtual Port Extender Quick Reference Guide*.

   Note

   If you are using unconfigured (`unconfigure switch all`) or "out-of-the box" controlling bridge switches (CBs), to ensure that full automation does not start to run, do not reboot the CBs after connecting the BPEs.

2. On the CB, enable VPEX mode. Use the command:

   ```
   enable vpex
   ```

   This command enables VPEX mode and allows you to refer to ports in the *slot*:*port* notation in applicable commands. A reboot is required for the command to take effect. After rebooting, the CLI prompt changes to show that the switch is now slot 1 (for example):

   ```
   Slot-1 VPEX 5520.14 #
   ```

3. (Optional) Set up a LAG with LACP on the switch to operate as the cascade port. Use the command:

   ```
   enable sharing port grouping port_list {algorithm [address-based {L2 | L3 | L3_L4 | custom} | }]} {resilient-hashing [on | off]} {distribution-mode [all | local-slot | port-lists]} {lacp | health-check}
   ```

   Note
   You cannot convert a cascade port to a LAG after the cascade port is created without first unconfiguring the slot from the cascade port.

4. Assign a slot number identifier to the attached BPE on the specified cascade port. Use the command:

   ```
   configure vpex ports port_list slot slot_num
   ```

   Note
   If the cascade port is part of a LAG, then configure the LAG master port as the VPEX port.

   The behavior of this command is similar to configuring slots within a chassis. After assigning a slot number to the BPE, you can make port-level configuration choices with the familiar *slot*:*port* notation in other commands involving the port extender's ports (for example, `configure vlan v1 add port 100:1`).

5. (Optional) Add a descriptive name to a BPE. Use the command:

   ```
   configure slot slot description [ slot_description | none ]
   ```

For information about troubleshooting an Extended Edge Switching topology, see Troubleshooting an Extended Edge Switching Topology on page 764.

Configuring Extended Edge Switching Ring Topology

Ensure that all BPEs are running the minimum VPEX version software to support Extended Edge Switching (see *ExtremeXOS Release Notes*). For information about upgrading the BPEs, see Upgrading the Controlling Bridge and Bridge Ports Extenders on page 782.
The following procedure explains how to set up an Extended Edge Switching ring topology.

Extended Edge Switching rings allow two Extended Edge Switching (VPEX) cascades to be joined together to form a ring, or a single cascade to be joined back to the controlling bridge (CB). When a link breaks or a bridge port extender (BPE) otherwise leaves, the remaining BPEs form one or two new cascades that end at the ring break, thus keeping connectivity to the CB alive.

For more information about Extended Edge Switching ring topology, see Extended Edge Switching Rings on page 742.

1. Connect the BPEs into one or two cascades.

2. Configure the cascades and allow them to form. You can use either:
   - Full automation (Configuring Extended Edge Switching Topology with Full Automation on page 757).
   - Partial automation (Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760).
   - Manual configuration (Manually Configuring Bridge Port Extenders on page 761).

   Note
   Ring re-balancing establishes two cascades of approximately equal length without regard to bandwidth consumption of a BPE. In a mixed node cascade, you should place the most bandwidth consuming BPEs as close as possible to the CB in the cascade. For example, a 48t BPE may use two-port LAGs to communicate with the CB, but a 24t BPE switch can only have one port in its upstream direction. Thus the 24-port BPEs should be placed at the tiers that are farther from the CB.

   Use the `show slot {slot {detail} | detail }` and `show vpex topology { port port_num} {summary | detail}` commands to ensure that the cascades have finished forming correctly.
3. Connect the end BPEs of these cascades to each other, or the end of the single cascade to a port on the CB.

   A message similar to the following appears:

   ```
   <Noti:VPEX.Ring.Dtect> Slot-1: Ring detected between ports <port1> and <port2>
   ```
4. (Optional) Turn on automatic ring re-balancing. Ring re-balancing is disabled by default. For more information about ring re-balancing, see Uneven Cascade Lengths and Ring Re-balancing on page 745. If you want to enable ring re-balancing, use the following command:

   `configure vpex ring rebalancing [auto | off]`

   When a ring contains different types of BPEs that provide different amounts of bandwidth, automatic ring re-balancing may cause an undesirable balance point as seen when using the `show vpex topology { port port_num} {summary | detail}` command. If so, wait until the ring state is complete. You can then force the balance point by disabling a cascade port at the desired re-balance point, disabling automatic re-balancing, and then enabling the cascade port that was disabled previously.
5. If you are using redundant CBs with one-arm MLAGs (see One-Arm MLAGs on page 746), have each MLAG peer declare a slot on the other peer's one-arm MLAG. Use the command:

   `configure vpex mlag-id mlag_id peer peer_name slot slot_num`
6. You can also set a preference for when Auto-Configuration configures the virtual MLAG ID. The default (cascade) configures the virtual MLAG ID whenever a 2nd-tier BPE is detected remotely, but not locally. To set the preference, use the command:

   `configure vpex auto-configuration mlag-id [cascade | ring]`

For information about troubleshooting an Extended Edge Switching topology, see Troubleshooting an Extended Edge Switching Topology on page 764.

Troubleshooting an Extended Edge Switching Topology

The following information provides tips on how to verify and troubleshoot your Extended Edge Switching topology.

Commands for Verifying/Troubleshooting Extended Edge Switching Topology

Use the following commands to verify and troubleshoot the Extended Edge Switching topology:

- `show configuration {module-name} {detail}` with *vpex* as the *module-name*—shows cascade port configuration.
- `show configuration {module-name} {detail}` with *vlan* as the *module-name* or with `| grep sharing`—shows LAG configuration.
- `show configuration {module-name} {detail}` with *devmgr* as the *module-name*—shows slot module type configuration.
- `show vpex`—shows mapping between cascade port and slot number.
- `show slot {slot {detail} | detail }`—shows status of BPE slot.
- `show vpex ports {ports_list}`—shows the status of the CSP control plane connections for each cascade port.
- `show vpex topology { port port_num} {summary | detail}`—shows the detected topology type, status, ring status (if applicable), and bridge port extender (BPE) interconnections.

Commands for Verifying/Troubleshooting Redundant Controlling Bridges (CBs)/MLAGs

To verify and troubleshooting the MLAG setup, use the following commands:

`show mlag peer {peer_name}`—shows if MLAG peer is up.

`show configuration {module-name} {detail}` with *vsm* as the *module-name*—shows MLAG port configuration.

Limitations When Using Automation to Set up Redundant CBs/MLAGs

The following limitations apply when using automation to set up redundant CBs.

Limitation	Workaround
If the CB has MLAG peer configuration, but its MLAG peer is down, you cannot enable auto-configuration.	• Remove the MLAG peer configuration to make it a single CB setup. • Bring up the MLAG peer.
If a BPE is configured before MLAG peer is configured, auto-configuration does not configure the MLAG port for the previously configured BPE.	Manually configure MLAG port for the previously configured BPE by setting the MLAG port ID to 5000 plus the BPE's slot number.

Selecting How VLAN Membership is Implemented for Extended Ports

In Extended Edge Switching hardware, the extended ports are represented as virtual ports. The VLAN membership of extended ports can be implemented in two ways:

- Hash table with VLAN and virtual port as key. Hash tables can lead to hash collisions at higher scale. (Default)
- Virtual port group. Programming the same group number in the VLAN table and virtual port table indicates membership. The hardware has 64 virtual port groups. You should select this option if many VLANs share the same extended ports.

Note that changing this configuration at run time could result in temporary loss of traffic while the tables are reprogrammed. It is preferable to identify which option works best for the particular topology and leave the setting unchanged during run time or schedule the change during a maintenance window.

To set the way VLAN membership is implemented, use the following command:

```
configure forwarding vpex vlan-port-filter [hash-table | port-group]
```

To see the current selection for VLAN membership, use the command:

```
show forwarding configuration
```

Configuring IP Multicast Replication

IP multicast (IPMC) replication involves generating multiple copies of incoming IPMC traffic to subscribed receivers. For more information about IPMC replication, see Optimized IP Multicast Replication on page 748.

To set the IPMC replication mode, on the controlling bridge or bridge port extenders, use the following command:

```
configure forwarding vpex ipmc replication [ controlling-bridge | bpe ]
```

To view the current selection for IPMC replication, use the following command:

```
show forwarding configuration
```

Unconfiguring VPEX Bridge Port Extenders

Unconfiguring the entire VPEX Setup and Disabling VPEX Mode

To unconfigure the entire VPEX setup on the switch and disable VPEX mode:

Disable VPEX mode. Use the command:

```
disable vpex
```

Disabling VPEX mode removes all port extender slot number assignments made using `configure vpex`. A reboot is required for this command to take effect.

Removing Bridge Port Extenders from Slot Assignments

You should only perform this procedure if you want to permanently remove the slot and all of its configuration, or if you want to rearrange the same BPEs into different slots.

To disassociate a bridge port extender (BPE) from a slot number assignment:

1. If the slot module is configured, use the command `unconfigure slot slot` .
2. `unconfigure` **`vpex ports`** *`port_list`* **`slot`**.

   The behavior of this command is similar to removing slots within a chassis. After executing this command, the BPE no longer occupies the slot number assignment.

Unconfiguring MLAG Slot Assignment

To unconfigure a slot previously declared on an , use the following command:

`unconfigure` **`vpex mlag-id`** *`mlag_id`* **`slot`**

Adding and Removing Bridge Port Extenders

The following information shows you how to add, replace, and remove bridge port extenders (BPEs) from cascades Extended Edge Switching topologies.

Replacing Bridge Port Extenders

You can replace bridge port extenders (BPEs) in existing Extended Edge Switching topologies.

This procedure explains how to replace a BPE with another BPE of the same model. If you want to replace an existing BPE with a different model, you need to remove the existing BPE, and then add the new BPE (see Inserting Additional Bridge Port Extenders on Cascades on page 766).

Note
For cascade topologies, control plane/traffic is disrupted on the cascade below the BPE being replaced until the new BPE is connected and accepted into the cascade.

To replace a BPE:

1. Detach the cables and remove the existing BPE.
2. Attach the cables to the new BPE.

Inserting Additional Bridge Port Extenders on Cascades

You can add additional bridge port extenders (BPEs) to existing Extended Edge Switching cascade topologies. You can add additional BPEs anywhere in the cascade. You can use either partial automation or manual configuration to add the additional BPE.

To add a BPE using **manual** configuration:

1. Choose the location of the BPE in the cascade:
   - *Not* at the bottom: Unconfigure all BPEs below the intended location for the new BPE in the cascade starting from the bottom, and then moving up, using the procedure in Removing Bridge Port Extenders from Slot Assignments on page 766.
   - *Bottom*: Complete step 2, and then you are done.
2. Attach the BPE, and configure it using the procedure in Manually Configuring Bridge Port Extenders on page 761 starting with step Manually Configuring Bridge Port Extenders on page 761.
3. Re-configure all of the BPEs below the new BPE starting from the top, and then moving down, using the same procedure as in step 2.

To add a BPE using **partial automation**, connect the BPE in the desired location, and then reboot the controlling (CB).

Inserting Additional Bridge Port Extenders on Rings

You can add bridge port extenders (BPEs) to existing Extended Edge Switching ring topologies. You can add BPEs anywhere in the ring.

To add a BPE:

1. Disconnect the cable between the two BPEs in the ring at the desired insertion point, which allows the ring to enter the severed state, and then re-converge into two operational cascades.
2. Connect the new BPE to the same connection points disconnected previously.

   If you have auto-configuration enabled (`enable` **`vpex auto-configuration`**), you are finished. If you want to manually configure the rest of the setup (ensure that auto-configuration is disabled), continue with the following steps.
3. (If you are connecting at the point where the configured cascades were joined, do *not* perform this step) Unconfigure the cascade port of the BPE connecting the new BPE using the procedure in Removing Bridge Port Extenders from Slot Assignments on page 766.

   Note
   To find where the configured cascades were joined, use the `show` **`vpex topology`** { **`port`** *`port_num`*} {**`summary`** | **`detail`**} command, and see where the dynamically created cascade ports ("d" flag) start on the cascade(s).

4. Configure the new BPE using the procedure in Manually Configuring Bridge Port Extenders on page 761 starting with step Manually Configuring Bridge Port Extenders on page 761.

Configuring a new VPEX port/slot in a ring destroys the ring, and changes the configured cascades that formed it. For the new cascades, one ends where the new slot/BPE was added and the other ends at the next BPE that is connected to the cascade with the new slot/BPE.

Removing Bridge Port Extenders from Cascades

You can permanently remove bridge port extenders (BPEs) from existing Extended Edge Switching cascades.

To remove a BPE:

1. Unconfigure the BPE to be removed, and all BPEs below the location of that BPE in the cascade starting from the bottom, and then moving up, using the command:

   `unconfigure` **`vpex ports`** *`port_list`* **`slot`**
2. Disconnect the desired BPE, and then re-cable the remaining BPEs as needed.
3. Re-configure all of the previously unconfigured BPEs, except for the removed BPE, starting from the top, and then moving down, using the procedure in Manually Configuring Bridge Port Extenders on page 761 starting with step Manually Configuring Bridge Port Extenders on page 761.
4. Unconfigure the removed BPE's slot assignment using the following command:

   `unconfigure slot` *`slot`*

To see an example of removing a BPE from a cascade, see Removing Bridge Port Extender from a Cascade Example on page 777.

Removing Bridge Port Extenders from Rings

You can permanently remove bridge port extenders (BPEs) from existing Extended Edge Switching rings.

To remove a BPE:

1. Disconnect both connections for the BPE. The ring enters the severed state.
2. Connect the two disconnected ports together. The ring does not re-form at this time.
3. Unconfigure the removed BPEs from the topology using the procedure in Removing Bridge Port Extenders from Slot Assignments on page 766.

   Configuring a new VPEX port/slot in a ring destroys the ring, and changes the configured cascades that formed it. For the new cascades, one ends where the new slot/BPE was added and the other ends at the next BPE that is connected to the cascade with the new slot/BPE.

Showing Bridge Port Extender Information

Use the following show commands to display information about the VPEX setup.

- To show the status of VPEX mode, and if applicable, switch port to bridge port extender (BPE) slot assignments, use the following command:

  `show vpex`
- To show status information about the BPE attached to the specified cascade ports, use the following command:

  `show` **`vpex ports`** {*`ports_list`*}

- To show BPE statistics control plane information on an individual per-cascade-port basis, use the following command:

  ```
  show vpex ports {port_list} {statistics} {detail}
  ```

- To show BPE Edge Control Protocol (ECP) counter information, use the following command:

  ```
  show vpex ports {port_list} ecp statistics
  ```

- To show attached BPE's cascade port, slot assignment number, and MAC addresses, use the following command:

  ```
  show vpex bpe
  ```

- To show power supply, temperature, and fan status for BPEs, use the following command:

  ```
  show vpex bpe {slot slot_num} {environment}
  ```

- To show general BPE control plane statistics, use the following command:

  ```
  show vpex bpe {slot slot_num} {statistics} {detail}
  ```

- To show CPU utilization information for BPEs, use the command:

  ```
  show vpex bpe {slot slot_num} cpu-utilization
  ```

- To show detailed image version information for BPEs, use the command:

  ```
  show vpex bpe { slot slot_num} version detail
  ```

- To show a BPE's name, use the following command:

  ```
  show slot {slot {detail} | detail }
  ```

- To show information about the Port Extension TLV, use the command:

  ```
  show lldp {port [all | port_list]} neighbors {detailed}
  ```

- To show status and interconnection details of attached BPEs, use the command:

  ```
  show vpex topology { port port_num} {summary | detail}
  ```

Taking a Bridge Port Extender Offline

You can take a bridge port extender (BPE) offline if needed. Note that in a cascade, disabling a cascade port takes the directly attached BPE offline with any downstream attached BPEs. When a BPE is taken offline, its extended ports go into a link down state.

In a cascade, to take a BPE offline, use the following command specifying the cascade port that the BPE is attached to in *port_list* (do not use the all option):

```
disable port [port_list | all]
```

Note
If the cascade port is part of a LAG, you must disable all ports in the LAG.

With a ring, disabling only the cascade or upstream port on a BPE results in a ring failover, so the BPE is not taken offline if only one connection is disabled. You must disable all ports that can connect the BPE back to the controlling bridge (CB). Also, if there are load sharing groups connecting the BPE to another BPE or CB, all ports in that group must be disabled just to sever one of the connections.

Rebooting VPEX

To reboot *all* attached BPEs, use the following command specifying the all option:

```
reboot {[time mon day year hour min sec] | cancel} {slot slot-number}
| node-address node-address | stack-topology {as-standby} | all} |
rolling}
```

To reboot a *single* BPE, use the preceding command, specifying the BPE's slot in *slot-number*.

Configuration Examples

Full Automation Configuration Example

The following example shows how to use full automation (see Configuring Extended Edge Switching Topology with Full Automation on page 757) to set up a controlling bridge (CB) with three attached bridge port extenders (BPEs) in a two-tier structure (see Figure 80 on page 774).

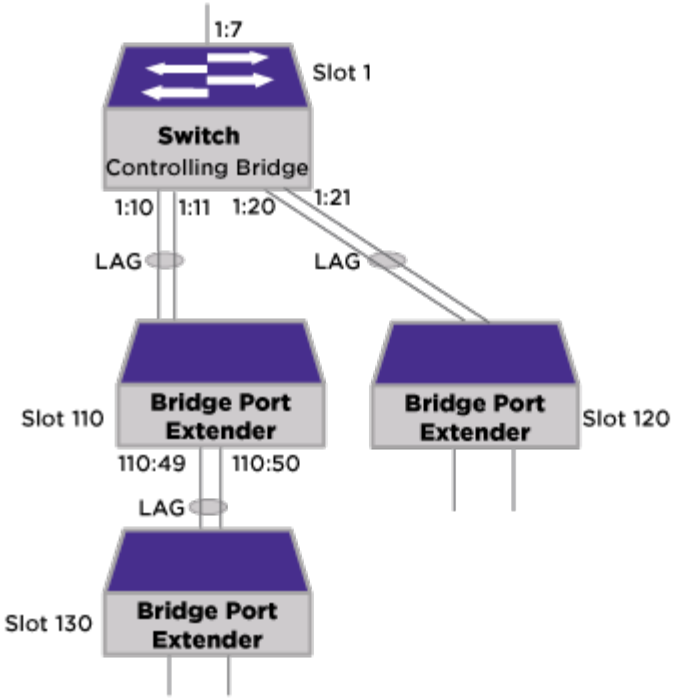


Figure 78: CB with Two-Tier BPEs Topology

1. Attach BPEs.
2. Check for the presence `default.xsf`:

```
# ls /usr/local/cfg
drwxrwxrwx    2 root     root          1024 Apr 10 15:01 cc_logs
drw-r--r--    2 root     root          1024 Dec  4 17:43 lost+found
-rw-r--r--    1 admin    admin       442088 Apr 29 18:05 primary.cfg
-rw-r--r--    1 admin    admin       451011 Mar 14 13:27 primary_nonslotted.cfg
-rw-r--r--    1 admin    admin      1256747 Feb 26 05:24 primary_slotted.cfg
drw-r--r--    4 root     root          1024 Feb 26 05:23 ssl
drwxr-xr-x    2 root     root          1024 May 22 14:58 vmt
```

3. Remove CB configuration:

```
# unconfigure switch all
Restore all factory defaults and reboot? (y/N) Y
```

   When the CB finishes rebooting, connected BPEs, extended slots, and LAGs on cascade ports are automatically configured.

4. Apply Layer 2 configuration:

   Note

   Due to unconfiguring the switch, a series of questions appear to apply the desired level of security configuration. Respond to the prompts as desired. For more information, see Using Safe Defaults Mode on page 48.

```
# configure vlan Default delete port all
# create vlan v100 tag 100
# configure vlan v100 add port 130:1-2, 120:1-2, 1:7 tagged
```

5. If required, you can use standard show commands to troubleshoot the network from the CB using the counters:

```
# show port 130:49-50, 120:49-50, 1:7 statistics no-refresh
Port   Link  Tx Pkt    Tx Byte    Rx Pkt    Rx Byte  Rx Pkt Rx Pkt  Tx Pkt Tx Pkt
       State  Count      Count     Count      Count   Bcast  Mcast   Bcast  Mcast
====== ===== ======= ========== ========= ========== ======= ====== ======= ======
130:1  A   97045942 6209457963         0          0       0      0       0     35
130:2  R          0          0         0          0       0      0       0      0
120:1  A         36       4715  97001731 6206623104       0      0       0     35
120:2  A   95802538 6897726741  95784987 7279063658       0    512       0    697
1:7     A   97045942       4715  97001731 6206623104       0      0       0      0
====== ===== ======== ========= ========= ========== ====== ====== ======= ======
          > in Port indicates Port Display Name truncated past 8 characters
          > in Count indicates value exceeds column width. Use 'wide' option or '0' to
clear.
          Link State: A-Active, R-Ready, NP-Port Not Present, L-Loopback

# show port 130:49-50, 120:49-50, 1:7 qosmonitor no-refresh
Port Qos Monitor
Port           QP1       QP2       QP3       QP4       QP5       QP6       QP7       QP8
               Pkt       Pkt       Pkt       Pkt       Pkt       Pkt       Pkt       Pkt
              Xmts      Xmts      Xmts      Xmts      Xmts      Xmts      Xmts      Xmts
================================================================================================
130:49   119682459         0         0         0         0         0         0        45
130:50           0         0         0         0         0         0         0         0
120:49           0         0         0         0         0         0         0        45
120:50   119803233         0         0         0         0         0         0       923
1:7      119803233         0         0         0         0         0         0       923
```

Partial Automation Configuration Example

The following example shows how to use partial automation (see Configuring Extended Edge Switching Topology with Partial Automation (Auto-configuration) on page 760) to set up a controlling bridge (CB) with three attached bridge port extenders (BPEs) in a two-tier structure (see Figure 80 on page 774).

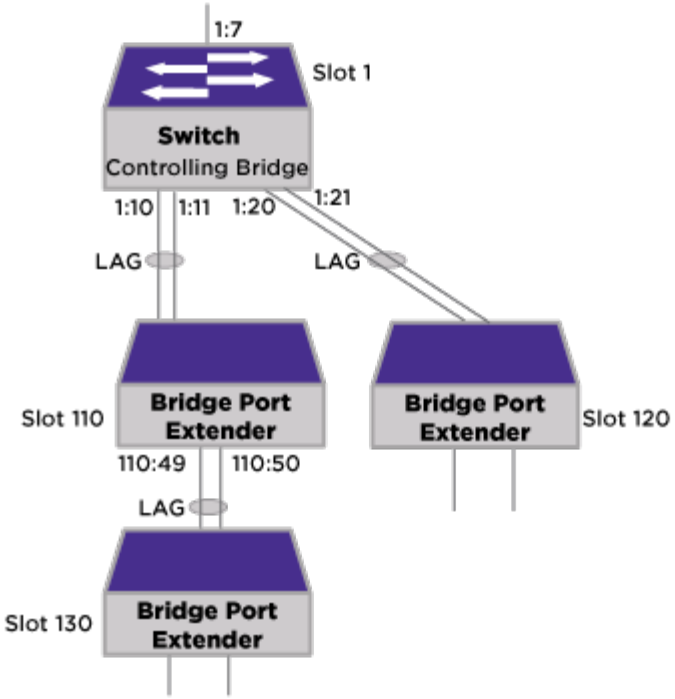


Figure 79: CB with Two-Tier BPEs Topology

1. Attach BPEs.
2. On the CB, enable VPEX mode:

```
465.1 # enable vpex
465.1 # reboot
Are you sure you want to reboot the switch? (y/N) Yes
Slot-1 VPEX 465.1 #
```

> **Note**
> After reboot, command prompt changes to show that you are in VPEX mode.

3. Enable auto-configuration:

```
Slot-1 VPEX 465.1 # enable vpex auto-configuration
VPEX Auto-Configuration is enabled for non-MLAG configuration.
```

4. Reboot the CB:

```
Slot-1 VPEX 465.1 # reboot
```

   When the CB finishes rebooting, connected BPEs, extended slots, and LAGs on cascade ports are automatically configured.

5. Apply Layer 2 configuration:

```
Slot-1 VPEX 465.1 # configure vlan Default delete port all
Slot-1 VPEX 465.2 # create vlan v100 tag 100
Slot-1 VPEX 465.3 # configure vlan v100 add port 130:1-2, 120:1-2, 1:7 tagged
```

6. If required, you can use standard show commands to troubleshoot the network from the CB using the counters:

```
Slot-1 VPEX 465.3 # show port 130:49-50, 120:49-50, 1:7 statistics no-refresh
Port   Link  Tx Pkt    Tx Byte    Rx Pkt    Rx Byte  Rx Pkt Rx Pkt  Tx Pkt Tx Pkt
```

```
       State  Count      Count      Count      Count    Bcast  Mcast   Bcast  Mcast
====== ===== ======= ========== ========== ========== ======= ====== ======= ======
130:1  A    97045942 6209457963          0          0       0      0       0     35
130:2  R           0          0          0          0       0      0       0      0
120:1  A          36       4715   97001731 6206623104       0      0       0     35
120:2  A    95802538 6897726741   95784987 7279063658       0    512       0    697
1:7     A    97045942       4715   97001731 6206623104       0      0       0      0
====== ===== ======== ========== ========== ========== ====== ====== ======= ======
          > in Port indicates Port Display Name truncated past 8 characters
          > in Count indicates value exceeds column width. Use 'wide' option or '0' to
clear.
          Link State: A-Active, R-Ready, NP-Port Not Present, L-Loopback

Slot-1 VPEX 465.13 # show port 130:49-50, 120:49-50, 1:7 qosmonitor no-refresh
Port Qos Monitor
Port           QP1       QP2       QP3       QP4       QP5       QP6       QP7       QP8
               Pkt       Pkt       Pkt       Pkt       Pkt       Pkt       Pkt       Pkt
              Xmts      Xmts      Xmts      Xmts      Xmts      Xmts      Xmts      Xmts
================================================================================================
130:49   119682459         0         0         0         0         0         0        45
130:50           0         0         0         0         0         0         0         0
120:49           0         0         0         0         0         0         0        45
120:50   119803233         0         0         0         0         0         0       923
1:7      119803233         0         0         0         0         0         0       923
```

Manual Configuration Example

The following example shows how to manually set up a controlling bridge (CB) with three attached bridge port extenders (BPEs) in a two-tier structure (see Figure 80).

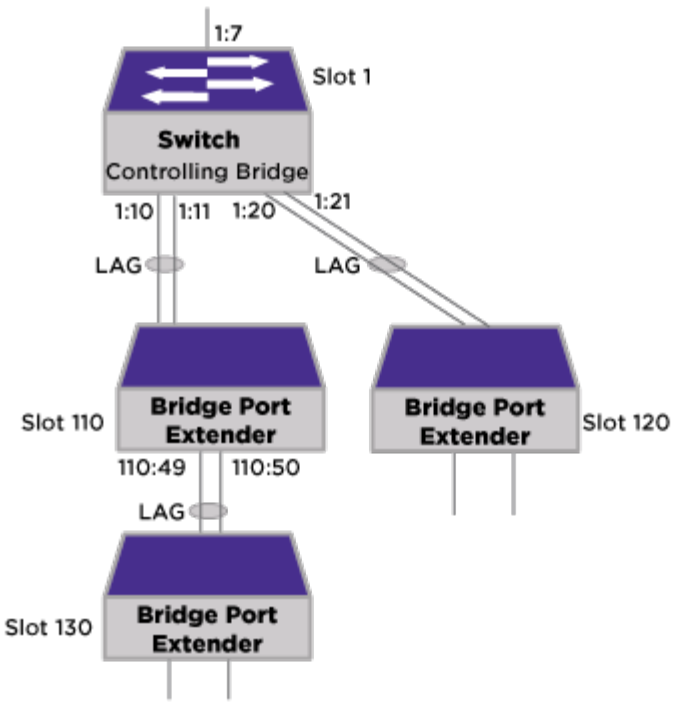


Figure 80: CB with Two-Tier BPEs Topology

1. Attach BPEs.

2. On the CB, enable VPEX mode:

```
5520.1 # enable vpex
5520.1 # reboot
Are you sure you want to reboot the switch? (y/N) Yes
Slot-1 VPEX 5520.1 #
```

Note
After reboot, command prompt changes to show that you are in VPEX mode.

3. Create LAGs between CB and first-tier BPEs:

```
Slot-1 VPEX 5520.1 # enable sharing 1:10 group 1:10-1:11 lacp
Slot-1 VPEX 5520.2 # enable sharing 1:20 group 1:20-1:21 lacp
```

4. Configure slot identifiers associated with the first-tier BPEs:

```
Slot-1 VPEX 5520.1 # configure vpex port 1:10 slot 110
WARNING: This command will remove VLAN membership from the
         port 1:10.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.2 # configure vpex port 1:20 slot 120
WARNING: This command will remove VLAN membership from the
         port 1:20.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.3 # show slot
Slots    Type                 Configured           State       Ports   Flags
-------------------------------------------------------------------------------
Slot-1   5520        5520        Operational   64    M
Slot-110 V400-48p-10GE4                           Operational   52
Slot-120 V400-48p-10GE4                           Operational   52
Flags : M - Backplane link to Master is Active
        B - Backplane link to Backup is also Active
        D - Slot Disabled
        I - Insufficient Power (refer to "show power budget")

Slot-1 VPEX 5520.9 # show vpex
Virtual Port Extender: Enabled
Cascade
Port     Slot
=============
1:10     110
1:20     120
```

Note
Observe that the `show slot` command shows the CB as slot 1 with the two BPEs as slots 110 and 120.

Note
Observe that the `show vpex` command shows that ports 1:10 and 1:20 are the cascade ports connecting to the BPEs.

5. Create LAG between first-tier and second-tier BPEs:

```
Slot-1 VPEX 5520.1 # enable sharing 110:49 group 110:49, 110:50 lacp
```

6. Configure slot identifier associated with the second-tier BPE:

```
Slot-1 VPEX 5520.1 # configure vpex port 110:49 slot 130
WARNING: This command will remove VLAN membership from the
```

```
         port 110:49.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.3 # show slot
Slots    Type                Configured          State       Ports   Flags
-------------------------------------------------------------------------------
Slot-1   5520        5520         Operational   64    M
Slot-110 V400-48p-10GE4                          Operational   52
Slot-120 V400-48p-10GE4                          Operational   52
Slot-130 V400-48p-10GE4                          Operational   52
Flags : M - Backplane link to Master is Active
        B - Backplane link to Backup is also Active
        D - Slot Disabled
        I - Insufficient Power (refer to "show power budget")

Slot-1 VPEX 5520.9 # show vpex
Virtual Port Extender: Enabled
Cascade
Port     Slot
=============
1:10     110
1:20     120
110:49   130
```

7. Apply Layer 2 configuration:

```
Slot-1 VPEX 5520.1 # configure vlan Default delete port all
Slot-1 VPEX 5520.2 # create vlan v100 tag 100
Slot-1 VPEX 5520.3 # configure vlan v100 add port 130:1-2, 120:1-2, 1:7 tagged
```

8. If required, you can use standard show commands to troubleshoot the network from the CB using the counters:

```
Slot-1 VPEX 5520.3 # show port 130:49-50, 120:49-50, 1:7 statistics no-refresh
Port   Link  Tx Pkt    Tx Byte    Rx Pkt    Rx Byte  Rx Pkt Rx Pkt  Tx Pkt Tx Pkt
       State  Count      Count     Count      Count   Bcast  Mcast   Bcast  Mcast
====== ===== ======= ========== ========= ========== ======= ====== ======= ======
130:1  A   97045942 6209457963         0          0       0      0       0     35
130:2  R          0          0         0          0       0      0       0      0
120:1  A         36       4715  97001731 6206623104       0      0       0     35
120:2  A   95802538 6897726741  95784987 7279063658       0    512       0    697
1:7     A   97045942       4715  97001731 6206623104       0      0       0      0
====== ===== ======== ========= ========= ========== ====== ====== ======= ======
          > in Port indicates Port Display Name truncated past 8 characters
          > in Count indicates value exceeds column width. Use 'wide' option or '0' to
clear.
          Link State: A-Active, R-Ready, NP-Port Not Present, L-Loopback

Slot-1 VPEX 5520.13 # show port 130:49-50, 120:49-50, 1:7 qosmonitor no-refresh
Port Qos Monitor
Port           QP1       QP2       QP3       QP4       QP5       QP6       QP7       QP8
               Pkt       Pkt       Pkt       Pkt       Pkt       Pkt       Pkt       Pkt
              Xmts      Xmts      Xmts      Xmts      Xmts      Xmts      Xmts      Xmts
================================================================================================
130:49   119682459         0         0         0         0         0         0        45
130:50           0         0         0         0         0         0         0         0
120:49           0         0         0         0         0         0         0        45
120:50   119803233         0         0         0         0         0         0       923
1:7      119803233         0         0         0         0         0         0       923
```

Removing Bridge Port Extender from a Cascade Example

The following example shows how to remove a bridge port extender (BPE) from a cascade.

This example removes the BPE at slot 101 in a four-tier cascade and connects the BPE at slot 100 to the BPE at slot 102 (see Figure 81).

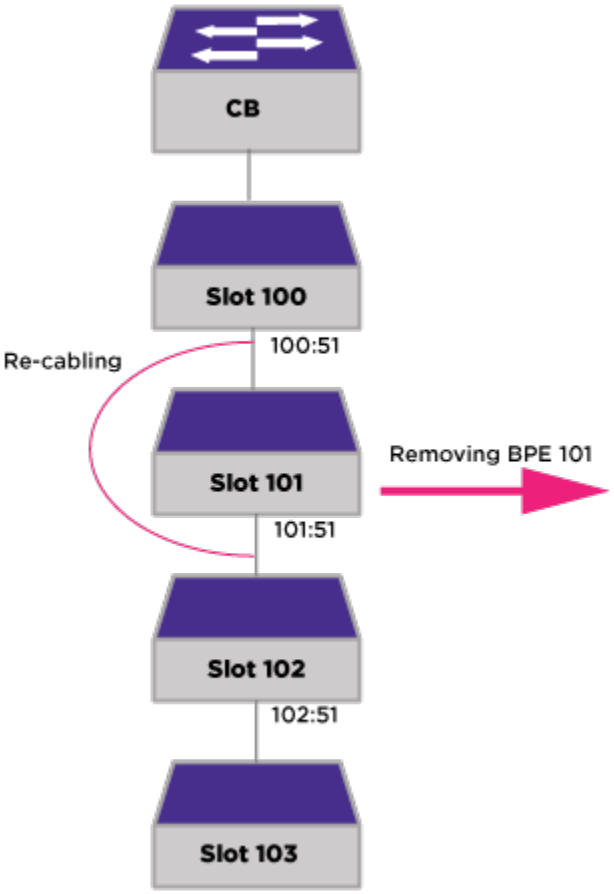


Figure 81: Removing BPE from Cascade

1. Unconfigure the bottom BPE:

```
# unconfigure vpex port 102:51 slot
```

2. Unconfigure the next BPE from the bottom:

```
# unconfigure vpex port 101:51 slot
```

3. Unconfigure the BPE being removed:

```
# unconfigure vpex port 100:51 slot
```

4. Disconnect BPE 101, and connect BPE 100 to BPE 102.
5. Re-configure the BPE at slot 100:

```
# configure vpex port 100:51 slot 102
```

6. Re-configure the BPE at slot 102:

```
# configure vpex port 102:51 slot 103
```

7. Remove the slot assignment (101) from the removed BPE:

```
# unconfigure slot 101
```

Redundant Controlling Bridges Configuration Example

The following example shows how to use full automation, partial automation, or manual configuration to set up a redundant controlling bridge (CB) topology with three attached bridge port extenders (BPEs) in a two-tier structure (see Figure 82).

For more information about redundant CBs, see Redundant Controlling Bridges on page 740.

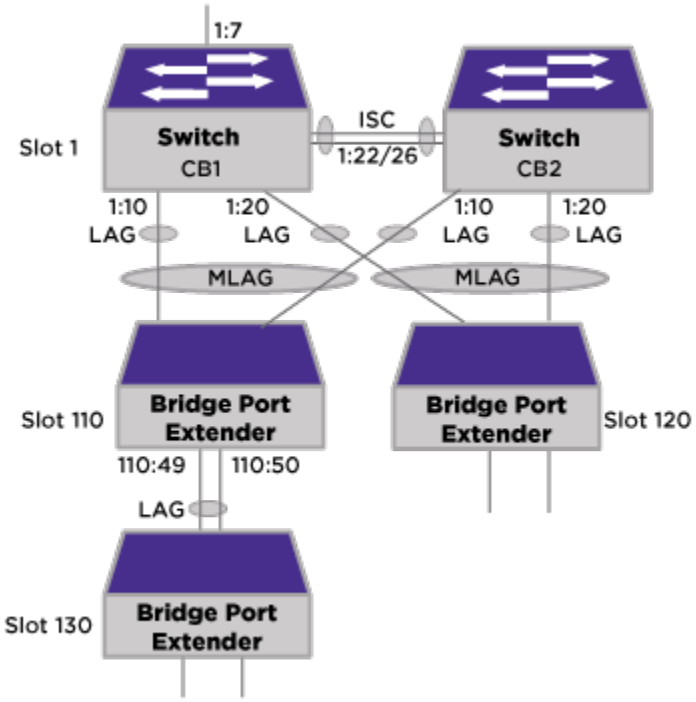


Figure 82: Redundant CBs with Two-Tier BPEs Topology

1. Connect the CBs.
2. Decide whether to use full or partial automation, or manual setup to configure the Extended Edge Switching topology:
   - For full automation, go to Full Automation Configuration Example on page 770.
   - For partial automation, go to Partial Automation Configuration Example on page 772.
   - For manual setup, continue with the rest of this procedure.
3. Create the peer relationship between the CBs on CB1:

```
enable sharing 1:22 group 1:22,1:26 lacp
create vlan isc tag 100
configure isc ipaddress 1.1.1.1/24
configure isc add port 1:22 tagged
```

```
create mlag peer vpex_peer
configure mlag peer vpex_peer ipaddress 1.1.1.2
```

4. Create the peer relationship between the CBs on CB2:

```
enable sharing 1:22 group 1:22,1:26 lacp
create vlan isc tag 100
configure isc ipaddress 1.1.1.2/24
configure isc add port 1:22 tagged
create mlag peer vpex_peer
configure mlag peer vpex_peer ipaddress 1.1.1.1
```

5. Start MLAG orchestration on CB1:

```
start orchestration mlag vpex_peer
```

Note
The CLI prompt changes to inform you that you are in orchestration mode.

Using orchestration mode enforces that all configuration commands are now checkpointed to the MLAG peer switch.

6. Create LAGs between CB and first-tier BPEs by substituting the following commands in step 3:

```
Slot-1 VPEX 5520.1 # enable sharing 1:10 group 1:10 lacp
Slot-1 VPEX 5520.2 # enable sharing 1:20 group 1:20 lacp
```

7. Set up the rest of the MLAG by executing the commands in the Manual Configuration Example on page 774, inserting the following commands between step 2 and 3:

```
enable mlag port 1:10 peer vpex_peer id 101
enable mlag port 1:20 peer vpex_peer id 102
```

When LACP shared ports are configured as MLAG ports, a LAG ID change after MLAG peer reboot may result in MLAG ports being removed and re-added to the aggregator.

8. To avoid the MLAG port flap, configure a common LACP MAC in both MLAG peers:

```
configure mlag peer peer_name lacp-mac lacp_mac_address
```

If the ISC port belongs to a higher port number and the MLAG port comes up before the ISC port on reboot/failover, then it might create a short loop.

9. To avoid the short loop, configure the reload delay timer for the MLAG ports to bring up the MLAG port with some delay:

```
configure mlag ports reload-delay <time>
enable mlag port reload-delay
```

The ISC port now comes up before MLAG port:

10. Enable MLAG port isolation to prevent packet forwarding over the ISC before the blocking filters are installed:

```
configure mlag ports link-up-isolation on
```

For MLAG best practices with VPEX, see the following article: *MLAG best practices with VPEX*

Simple Ring Configuration Example

The following example shows how to set up a simple Extended Edge Switching ring configuration (see Figure 83).

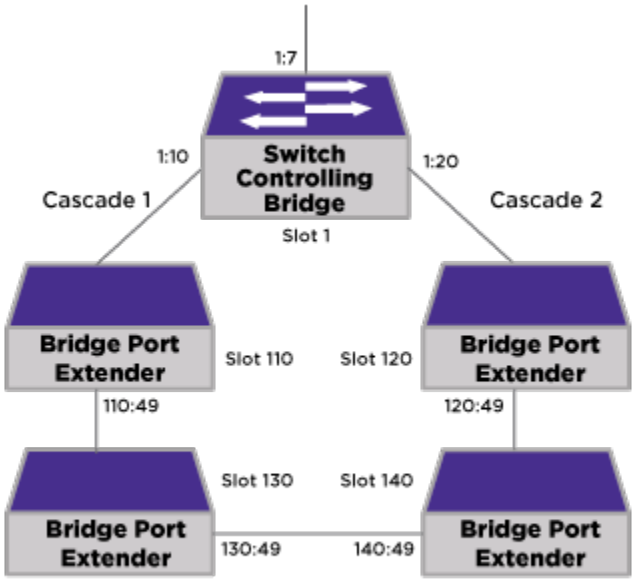


Figure 83: CB with Simple Ring Topology

1. Attach BPEs.
2. On the CB, enable VPEX mode:

```
5520.1 # enable vpex
5520.1 # reboot
Are you sure you want to reboot the switch? (y/N) Yes
Slot-1 VPEX 5520.1 #
```

Note
After reboot, command prompt changes to show that you are in VPEX mode.

3. Configure slot identifiers associated with the first-tier BPEs in cascades 1 and 2:

```
Slot-1 VPEX 5520.1 # configure vpex port 1:10 slot 110
WARNING: This command will remove VLAN membership from the
         port 1:10.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.2 # configure vpex port 1:20 slot 120
WARNING: This command will remove VLAN membership from the
         port 1:20.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.3 # show slot
Slots    Type                Configured          State       Ports   Flags
-------------------------------------------------------------------------------
```

```
Slot-1   5520        5520        Operational   64    M
Slot-110 V400-48p-10GE4                           Operational   52
Slot-120 V400-48p-10GE4                           Operational   52
Flags : M - Backplane link to Master is Active
        B - Backplane link to Backup is also Active
        D - Slot Disabled
        I - Insufficient Power (refer to "show power budget")

Slot-1 VPEX 5520.9 # show vpex
Virtual Port Extender: Enabled
Cascade
Port     Slot
=============
1:10     110
1:20     120
```

Note

Observe that the `show slot` command shows the CB as slot 1 with the two BPEs as slots 110 and 120.

Note

Observe that the `show vpex` command shows that ports 1:10 and 1:20 are the cascade ports connecting to the BPEs.

4. Configure slot identifiers associated with the second-tier BPEs in cascades 1 and 2:

```
Slot-1 VPEX 5520.1 # configure vpex port 110:49 slot 130
WARNING: This command will remove VLAN membership from the
         port 110:49.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.2 # configure vpex port 120:49 slot 140
WARNING: This command will remove VLAN membership from the
         port 120:49.
         Do you want to continue? (y/N)Yes

Slot-1 VPEX 5520.3 # show slot
Slots    Type                Configured          State       Ports   Flags
-------------------------------------------------------------------------------
Slot-1   5520        5520        Operational   64    M
Slot-110 V400-48p-10GE4                           Operational   52
Slot-120 V400-48p-10GE4                           Operational   52
Slot-130 V400-48p-10GE4                           Operational   52
Slot-140 V400-48p-10GE4                           Operational   52
Flags : M - Backplane link to Master is Active
        B - Backplane link to Backup is also Active
        D - Slot Disabled
        I - Insufficient Power (refer to "show power budget")

Slot-1 VPEX 5520.9 # show vpex
Virtual Port Extender: Enabled
Cascade
Port     Slot
=============
1:10     110
1:20     120
```

```
110:49   130
120:49   140
```

Note
Observe that the `show slot` command shows the CB as slot 1 with the four BPEs as slots 110, 120, 130, and 140.

Note
Observe that the `show vpex` command shows that ports 1:10, 1:20, 110:49, and 120:49 are the cascade ports connecting to the BPEs.

5. Connect the end of cascade 1 to cascade 2 to form the ring:

```
<Noti:VPEX.Ring.Dtect> Slot-1: Ring detected between ports 140:49 and 130:49
```

Upgrading the Controlling Bridge and Bridge Ports Extenders

Periodically, there will be software upgrades for the controlling bridge (CB) and bridge ports extenders (BPEs).

The CBs run ExtremeXOS software, while the BPEs run VPEX software that is either packaged with the ExtremeXOS software or provided as a separate modification file. The different image files are distinguishable by their file extension:

- .xos—ExtremeXOS CB-only image file (example: SummitX-22.6.0.19.xos).
- .xmod—ExtremeXOS modification file with VPEX software for the BPEs (example: SummitX-22.6.0.19-vpex-bpe-1.1.0.41.xmod).

  Note
  Starting with ExtremeXOS 30.2, the diagnostic image is in a separate xmod file (example: onie-30.2.2.x-vpex_bpe_diag-1.0.0.1.xmod) that is not included in the regular BPE xmod or combined list files, but is installed on the BPE at the factory. We anticipate that you will not generally need to update this image.

- .lst—Combined ExtremeXOS CB software and VPEX BPE file (example: summitX-22.6.0.19-vpex_controlling_bridge.lst).

For complete information about how to perform a software upgrade, see the Software Upgrade and Boot Options on page 1697 chapter.

There are two methods for upgrading BPEs:

- **Automatic**—The advantage of automatically upgrading BPEs is that it is easier than manual upgrading, and does not require manual intervention.
- **Manual**—You can upgrade devices individually at a time of your choosing. The disadvantage is that you must initiate the upgrade manually on each CB and BPE.

Automatic Upgrading

Upgrading automatically uses .lst files, so that the CB and BPEs are upgraded together to ensure that all hardware is up-to-date.

When the upgrade is performed, the CB is upgraded (requires a reboot), and when the BPE boots up, it performs a version check. If the BPE image is up-to-date, the BPE progresses to the operational state. However, if the BPE image needs to be upgraded, the BPE retrieves the software image file from the CB, upgrades itself, and then reboots.

Starting with ExtremeXOS 30.5, an improved version of automatic upgrading is available for MLAG/redundant CB configurations. You can reboot each CB sequentially, rather than rebooting both simultaneously, avoiding the loss of network connectivity. To use automatic upgrading, ensure that:

- Both CBs are running the same ExtremeXOS and BPE xmod version.
- ISC connectivity is present (required for information exchange between CBs during upgrade).

1. Ensure that automatic upgrading is enabled (default behavior). Use the `show vpex` command to see if automatic upgrading is enabled. If not enabled, use the `enable` **`vpex auto-upgrade`** command to enable it.
2. On the first CB, install the .lst image. After installation, reboot the first CB.
3. After the first CB has finished rebooting and has synchronized all of the slot and ring information, on the second CB, install the.lst image, and then reboot it.

Note
If you do not have automatic upgrading enabled when using redundant CBs, and a BPE image upgrade is needed, you must reboot both CBs simultaneously. Simultaneous reboot is required because sequential rebooting (waiting for each to come fully operational before proceeding to reboot the other) maintains BPE connectivity. Because the BPE does not re-attach, it is not upgraded if needed.

After an upgrade occurs, the log shows messages similar to the following indicating that the BPE has been upgraded:

```
08/04/2017 17:57:25.69 <Info:HAL.Card.Info> Slot-1: VPEX BPE slot 100 Active image VER
1.1.0.41
08/04/2017 17:53:44.52 <Info:HAL.Card.Info> Slot-1: Upgrading VPEX BPE slot 110 image
from ver 1.1.0.40 to 1.1.0.41
08/04/2017 17:53:43.70 <Info:HAL.Card.Info> Slot-1: VPEX BPE slot 120 Active image VER
1.1.0.40
```

Manual Upgrading

For manual upgrading, you update each CB and BPE image individually in a “rolling” fashion.

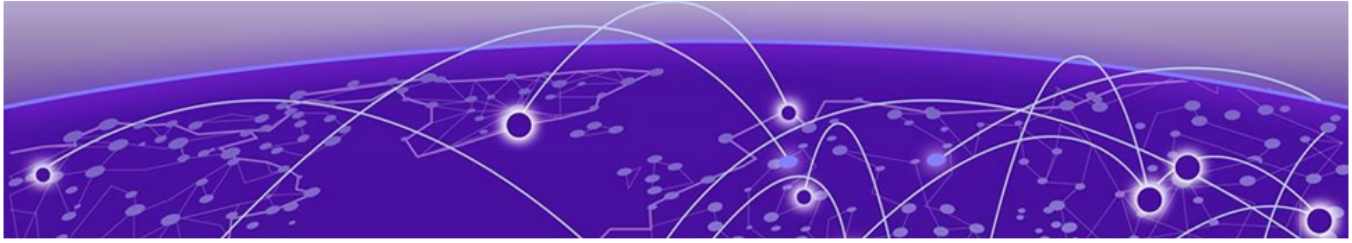
To manually upgrade:

1. Upgrade ExtremeXOS on your first CB using .xos image file (install image, and then reboot); do not install xmod.
2. For redundant CB topologies, repeat 1 for each redundant CB.

3. Install the xmod on each CB, and then reboot each BPE one at a time, using `reboot slot` *`BPE slot`* on the master CB, allowing the upgrade and initialization to complete for each BPE before proceeding to the next.

   Note
   To determine which CB is the master, use the `show` **`vpex ports`** {*`ports_list`*} command. The switch with the M flag is the master CB.

   Note
   Rebooting a BPE tears down the connection to any additional BPEs that are cascaded from that BPE, so proper redundancy planning is required.



Fabric Attach

Fabric Attach Overview on page 785
Configuring Fabric Attach on page 799
Fabric Attach Timeout Configurations on page 802
Fabric Attach and Extended Edge Switching Configuration Examples on page 803

The Fabric Attach feature allows attachment of non-SPB (Shortest Path Bridging) stations to an SPB network through a Fabric Attach server or it can act as a Fabric Attach server for *VXLAN (Virtual Extensible LAN)* overlay networks.

Fabric Attach Overview

Fabric Attach uses the IEEE802.1ab *LLDP (Link Layer Discovery Protocol)* extensions to automatically attach network devices to individual services in a IEEE 802.1aq Shortest Path Bridging (SPB) network. These network devices typically do not support SPB, MAC-in-MAC (802.1ah) or Network Services Identifier (NSI)/Individual Service Identifier (ISID) usage, and therefore cannot easily take advantage of the Fabric infrastructure without manual configuration of VLAN attachments to NSIs or ISIDs in multiple locations. Fabric Attach deals with this issue by facilitating automated network device discovery and the automatic configuration and teardown of NSI/ISID to VLAN associations at the edge of the network. Fabric Attach can also use the same LLDP extensions to allow Fabric Attach clients and proxy devices to connect to *VXLAN* overlay networks.

Note
Both terms NSI and ISID are used in this document, with NSI being the preferred term. For ExtremeXOS, the NSI is an ISID in SPB fabrics, but the NSI could also be used to as an identifier for other (non-SPB) fabrics where it will not be an ISID.

The feature includes conditional logic for automatic fallback management VLAN creation that prevents disruption in environments with existing VLAN configurations while preserving zero-touch provisioning automation.

Fabric Attach supports the following:

- Single *LAC (Link Aggregation Group)* between any two devices
- SPBM networks: Fabric Attach client and proxy behavior
- VXLAN overlay networks: Fabric Attach client, proxy, and server behavior
- Connection to a single Fabric Attach server

- One-to-one mapping between a VLAN and an NSI/ISID
- Stacking
- *RADIUS (Remote Authentication Dial In User Service)* authentication

Limitations

Mappings of VLANs to NSIs/ISIDs is limited to 94.

Fabric Attach does not support *MIB (Management Information Base)*s.

Fabric Attach Operation Modes

Fabric Attach has four modes of operation:

- Proxy mode—see Fabric Attach Proxy Mode on page 786.
- Server mode—see Fabric Attach Server Mode on page 786.
- Standalone proxy mode—see Fabric Attach Standalone Proxy Mode.
- Client mode—see Fabric Attach Client Mode.

Fabric Attach Proxy Mode

After the Fabric Attach proxy receives a Fabric Attach Element TLV from in an *LLDP* frame from a Fabric Attach server, the Fabric Attach Element TLV is included in all LLDP frames sent by the Fabric Attach proxy. The Fabric Attach Assignment TLV is included in LLDP frames to the Fabric Attach server if any VLAN to NSI mappings exist. The Fabric Attach Assignment TLV is included in LLDP frames sent to a Fabric Attach client if that client has requested at least one VLAN to NSI mapping. Mappings that are rejected by the server are placed in the “rejected” state in the LLDP database and remain in the database until they expire.

Supported Platforms

All platforms.

Fabric Attach Server Mode

Fabric Attach operates in server mode whenever BGP Auto-peering is configured or an LTEP (Local Tunnel End Point) is created. The Fabric Attach server may receive mapping requests from any of several sources (CLI, client/proxy requests using *LLDP*, BGP, OSPFv2, or network authentication services such as *RADIUS* and policy). Since the CLI prevents conflicting CLI-sourced mappings from being requested, a mapping request from the CLI is entered into the database as “accepted” and conflicting mappings in the database have their status set to “rejected”. When a mapping is requested by a client or proxy, it is entered into the database.

Supported Platforms

All platforms.

Fabric Attach Standalone Proxy Mode

Fabric Attach standalone proxy allows for Fabric Attach proxy functionality in environments without a Fabric Attach server.

Fabric Attach standalone proxy operation supports standard Fabric Attach proxy processing as if a Fabric Attach server has been discovered. You can enable or disable Fabric Attach standalone proxy support (see Configuring Fabric Attach). By default, it is disabled. Enabling the Fabric Attach Standalone Proxy mode enables immediate processing of pending NSI/VLAN bindings, if other configuration data, such as static uplink data allows the processing. Previously established settings based on Fabric Attach proxy operation, if present, are reset when Fabric Attach standalone proxy operation is enabled.

The Fabric Attach standalone proxy does not send provisioning requests upstream. A Fabric Attach standalone proxy automatically accepts requests from Fabric Attach clients and assumes that the upstream network has been provisioned appropriately. Disabling Fabric Attach standalone proxy mode resets configured NSI/VLAN binding data to its default state and enables full Fabric Attach Proxy operation (see Fabric Attach Proxy Mode). In Fabric Attach standalone proxy mode, you must provide the Fabric Attach server uplink information, which is typically gathered through Fabric Attach server discovery. After you provide this information, Fabric Attach standalone proxy mode operates as if a Fabric Attach server has been discovered and is accepting NSI/VLAN binding requests. The binding clean-up is similar to a Fabric Attach server timeout event, and occurs when the static uplink is deleted and when Fabric Attach standalone proxy operation is disabled.

Supported Platforms

All platforms.

Fabric Attach Client Mode

After booting up, Fabric Attach defaults to operation as a client and identifies as the Fabric Attach client type "switch". In Fabric Attach, there are element types for Fabric Attach server (both auth and non-auth), proxy (both auth and non-auth), and an unused element type of "Other". The rest of the values are all different variations of an Fabric Attach client. As an ExtremeXOS Fabric Attach client, all Fabric Attach devices are ignored, except proxy and server. If a proxy is detected, the client treats it like a server and sends any assignments (mappings) that it has to the proxy for relay to the server. If a server is detected (while operating as a client), the Fabric Attach database is cleared and proxy mode is activated. If connection to the server is lost, the switch will again become a client after the server connection times out.

Supported Platforms

All platforms.

Network Service Identifier Mappings

A Network Service Identifier (NSI) or Individual Service Identifier (ISID) to VLAN mapping entered into the database remains until one of the following events occurs:

- The mapping times out (client-requested mappings only).
- Mapped VLAN is removed.
- Switch is rebooted.
- A mapping request with higher precedence is received.
- The port on which the mapping was requested is administratively disabled.

There are three ways to configure an NSI/ISID mapping: CLI, *LLDP*, or *RADIUS*/Policy/*NetLogin*. The Fabric Attach feature only allows a one-to-one mapping of NSI/ISID to VLAN. However, it is possible for a mapping made through the CLI to conflict with mappings requested by a Fabric Attach client or by RADIUS. Conflicting mappings from multiple sources are resolved in the following precedence order (highest to lowest):

1. CLI (see Configuring Fabric Attach on page 799).
2. LLDP client request. (see LLDP).
3. RADIUS/Policy/NetLogin (see RADIUS/Policy/NetLogin on page 789).
4. Static NSI Offset (see Static NSI Offset).

Note
When operating as a Fabric Attach server, this precedence order is not observed, only the last request is accepted.

This precedence is implemented by sending only the highest precedence mapping for each VLAN to the Fabric Attach server in the Fabric Attach Assignment TLV. If the highest precedence mapping changes, the prior highest precedence mapping is no longer be sent to the Fabric Attach server and it is replaced with the "new" highest priority mapping. The Fabric Attach server rejects the new mapping until the prior mapping times out of the Fabric Attach server's database.

When a VLAN to NSI/ISID mapping is requested by a Fabric Attach client, LLDP determines if the requested mapping conflicts with an existing client or CLI manual mapping. If this is the first client-requested mapping for a VLAN, it is added to the LLDP database with a “pending” status and is added to the list of mappings that are sent to the Fabric Attach server.

Conflicts between mapping requests from the same source are handled as follows: A conflict arises when different NSI values are requested for a VLAN. When a conflict arises between Fabric Attach client requests, the NSI with the lowest numerical value is retained, overwriting the previous client requested value for the VLAN. This reduces flip-flopping between competing values, since these mappings are continuously requested by Fabric Attach clients through LLDP. The conflict is resolved when all Fabric Attach clients have been configured to request the same mapping. After the incorrect mapping is no longer requested by any Fabric Attach client, it times out of the database and is replaced by the correct value from the next Fabric Attach client request. The default timeout value for the database on the FA proxy is 240 seconds and is configurable beginning with version 33.2.1. When a conflict arises because of

RADIUS requests, the last NSI value received is used, overwriting any previous NSI from RADIUS for this VLAN. After the RADIUS configuration has been corrected so that authentications return the correct NSI value, the correct mapping is then provided by the next user authentication. RADIUS can be configured to re-authenticate users periodically, but this is not required and if there is no change to the data for a user, RADIUS provides no notification. Since RADIUS does not provide periodic confirmation of the mappings it has requested, these mappings do not time out.

LLDP

LLDP may receive mapping requests from the CLI and from attached Fabric Attach clients. The CLI prevents you from configuring multiple static (CLI sourced) mappings for a VLAN. When a VLAN to NSI/ISID mapping is configured through the CLI (see Configuring Fabric Attach on page 754), the mapping is entered into the database and is persistent across device reboots, whereas mappings requested by Fabric Attach clients are not persistent.

RADIUS/Policy/NetLogin

RADIUS/Policy/NetLogin mappings are not persistent.

Actions when RADIUS is enabled::

- Policy maptable may be configured to use policy, tunnel, or both.
- RADIUS may return a policy name, a VLAN, and possibly an Network Service Identifier (NSI) mapping.

  If RADIUS returns a VLAN/NSI mapping:
  - Policy is **not** enabled: policy does not install an NSI mapping.
  - Policy **is** enabled: the mapping is installed if the following conditions are met.
    - Policy must be configured to make use of the RADIUS (RFC3580) VLAN, meaning:
      - The Policy maptable response must be set to “tunnel” or “both”.
      - If the response is set to “policy” (the default), the VLAN/NSI is not used.
      - Vlanauthorization must be enabled.
    - The VLAN and NSI must both be specified for the NSI to be used (see Defining VLAN/NSI Mappings with RADIUS Standards Attributes or VSAs on page 789).
    - The authentication response is an initial authentication for the given user or a re-authentication of an existing user with a mapping that differs from the existing mapping.

Defining VLAN/NSI Mappings with RADIUS Standards Attributes or VSAs

There are two ways to define the VLAN/NSI mapping using a combination of RADIUS Standards (*RFC2868* and *RFC3580*) Attributes and/or Vendor Specific Attributes (VSAs).

1. The presently supported *RFC3580* VLAN can be associated with newly introduced Extreme Networks VSAs.
   - RFC2868 & RFC3580 RADIUS Attributes:
     - Attribute 64: Tunnel-Type = VLAN (13)
     - Attribute 65: Tunnel-Medium-Type = 802

    - Attribute 81: Tunnel-Private-Group-Id=<VlanID>
  - Extreme Networks VSAs:
    - Attribute 230: Extreme-NSI-Type
    - Attribute 231: Extreme-NSI-ID

  If configuring these attributes manually, care must be taken. Extreme-NSI-Type and Extreme-NSI-ID values require a “tag” byte to allow for multiple attribute pairs to be specified in the same RADIUS response (For example, in FreeRADIUS this is annotated with “has_tag”). At present, only one Type/ID pair is used. If more than one pair is present, the entry with the lowest tag value is used. This is associated with the Tunnel-Private-Group-Id’s VLAN. Note that although the Tunnel-Private-Group-id attribute also supports an optional “tag” value as well, for backwards compatibility it is not currently used. Whether or not a “tag” value is specified in the Tunnel-Private-Group-Id attribute, it is matched to the Extreme-NSI-ID. Future releases may place additional restrictions on mismatched tags between the Tunnel-Private-Group-Id attribute and the Extreme-NSI-ID attribute.

2. Alternatively, an existing Nortel/Avaya attribute can also be used. The attribute is of the form “VLAN:NSI”. As the VLAN is specified within the attribute, *RFC2868* and *RFC3580* attributes are not required:
   - Attribute 171: Fabric-Attach-ISID

Note
The "policy maptable response" must be configured as both/tunnel for the feature to be effective, as the NSI mapping to egress VLANs is supported only using RADIUS. When configured as tunnel, RADIUS derived attributes are applied with the first valid VLAN:NSI being the PVID. When configured as policy, the existing behavior continues.

Note
If both attributes are present in the RADIUS attributes returned, the Extreme VSAs is used.

Note
A maximum of 94 VLAN:NSI pairs is processed per user. This is based on maximum VLAN:NSI values that can be sent by Fabric Attach in LLDP FA TLV.

Starting with ExtremeXOS 22.5, policy can configure NSI mappings based on the RADIUS-returned “policy name”. This allows the mappings to be derived from the RADIUS configuration and avoid configuration conflicts between users. This makes it easier for all users that match a policy profile to get the same mapping.

Note
Policy and RADIUS authentication is performed per-user, which means NSI mappings are also specified per user. Unless a common policy profile is used, you cannot prevent different users from mapping a VLAN to different NSI values.

Sample output for RADIUS attribute:

```
F06426700000 Cleartext-Password := "NOPASSWORD"
                  Filter-Id := "dacl",
                  Service-Type := Framed-User,
                  Session-Timeout := "600",
                  Termination-Action := RADIUS-Request,
                  Tunnel-Medium-Type := "IEEE-802",
                  Tunnel-Private-Group-Id := "324",
                  Tunnel-Type := "VLAN",

                  Fabric-Attach-VLAN-ISID :=
"1000:281111,1001:212,1020:213,1003:2400,1004:21110,1005:21111",
                  Fabric-Attach-VLAN-ISID +=
"1006:1006000,1007:1007,1008:1008,1009:1009,1110:1110,1111:11111"
```

Multiple Fabric Attach VLAN-ISID Attributes

Version 33.2.1 adds support for processing multiple VLAN/NSI pairs in Radius VSA 171: Fabric-Attach-ISID. The processed VLAN/NSI information is propagated using LLDP Fabric Attach TLV if a Fabric Attach Server is present.

Supported Platforms

All platforms.

Example

Example configuration for Multiple Fabric Attach-VLAN-ISID:

```
configure policy rule-model access-list
configure policy slices tci-overwrite 2
configure policy slices shared 2

configure netlogin port 3:32 allowed-users 768

configure policy profile 1 name "test" pvid-status "enable" pvid 22 tci-overwrite
"enable" nsi 121
configure policy profile 2 name "sqa" pvid-status "enable" pvid 23 tci-overwrite "enable"
nsi 1002
configure policy profile 3 name "policy" access-list "policy" pvid-status "enable" pvid
44 tci-overwrite "enable"
create policy access-list policy.ace matches tcpsourceportIP 34-50:1.1.1.1 mask 64
actions syslog forward cos 5
configure policy maptable response both
configure policy vlanauthorization enable
enable policy
enable netlogin mac
enable netlogin ports 3:32 mac
configure netlogin add mac-list 00:00:01:00:00:00 24 encrypted
"#$fSY+a7mwkRY7A2hb5qxTlNVojps+ig=="
configure netlogin add mac-list f0:64:26:70:00:00 32 encrypted
"#$QJZceerJZ9paDNqjwwLXKOzOzg6Ryw=="
configure netlogin add mac-list ff:ff:ff:ff:ff:ff 48
Slot-1 Stack.17 # show ports 3:32 vlan
         Untagged
Port     /Tagged   VLAN Name(s)
-------- --------  -------------------------------------------------------------
3:32     Untagged  SYS_VLAN_3111, v4091
         Tagged    SYS_VLAN_0100, SYS_VLAN_0101, SYS_VLAN_0102, SYS_VLAN_0103,
                   SYS_VLAN_0104, SYS_VLAN_0105, SYS_VLAN_0106, SYS_VLAN_0107,
                   SYS_VLAN_0108, SYS_VLAN_0109, SYS_VLAN_0110, SYS_VLAN_0302,
                   SYS_VLAN_2000, SYS_VLAN_2001, SYS_VLAN_3200, SYS_VLAN_3999,
```

```
                  v300

(CIT_33.2.0.134) Slot-1 Stack.19 # show fabric attach assignments
Fabric Attach Mode: Proxy
ISID/NSI Offset   : None
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
          111  SYS_VLAN_0111                     Dynamic       111  Active
          112  SYS_VLAN_0112                     Dynamic       112  Active
          113  SYS_VLAN_0113                     Dynamic       113  Active
          114  SYS_VLAN_0114                     Dynamic       114  Active
          115  SYS_VLAN_0115                     Dynamic       115  Active
          116  SYS_VLAN_0116                     Dynamic       216  Active
          117  SYS_VLAN_0117                     Dynamic       117  Active
          300  v300                              Dynamic       301  Active
          302  SYS_VLAN_0302                     Dynamic     14444  Active
         2000  SYS_VLAN_2000                     Dynamic      2000  Active
         2001  SYS_VLAN_2001                     Dynamic      2001  Active
         3200  SYS_VLAN_3200                     Dynamic       300  Active
         3999  SYS_VLAN_3999                     Dynamic       300  Active
         4091  v4091                             Dynamic         1  Active
```

Fabric Attach Automatic LAG Creation

When an switch receives Fabric Attach element TLVs on two or more ports that have same System-Id, same SMLT-Id, and have a Connection Type of SMLT, it automatically creates a LAG consisting of those ports. These LAGs:

- Do not persist after a reboot—after a reboot, TLVs must be received again to re-form the LAG.
- Do not require receipt of LACP PDUs—LACP PDUs are not sent.
- Require at least two ports to form the LAG— When a LAG is initially created with two ports, the lowest numbered port is chosen as the master port of the LAG. If there are three or more ports on the LAG, and the port is not the master port, it is simply removed from the LAG. If it is the master port, the lag is torn down, and then re-created with the lowest port number of the remaining ports as the master port.
- Are supported in stacks—the dynamic LAG is checkpointed to support failover, etc.

A port is removed from a LAG under following circumstances:

- Fabric attach is disabled using the `configure fabric attach ports` *`port`* `disable` command. If the port is active, then the next LLDP packet processing removes the port form a LAG. If the port is not active, it will be removed immediately from a LAG.
- TLV with Connection Type other than MLT is received.
- TLV with a System-ID that is different from the one when the LAG was created is received.
- TLV with a SMLT-ID that is different from the one when the LAG was created is received.

Note
The last three situations indicate that you are not connected to same Fabric Attach server, or the same LAG on a Fabric Attach server, which are not valid scenarios.

Note
MLAG with Fabric Attach Automatic LAG is not supported in ExtremeXOS 31.1 and later. Do not enable MLAGs on ports where automatically created LAGs will be created.

To see a configuration example, see Fabric Attach Automatic LAG Configuration Example on page 823.

Fabric Attach Support over Multi-switch Link Aggregation Groups (MLAGs)

Starting with ExtremeXOS 30.2, Fabric Attach server and proxy can work over . Starting with ExtremeXOS 30.4, Fabric Attach standalone proxy works over MLAG.

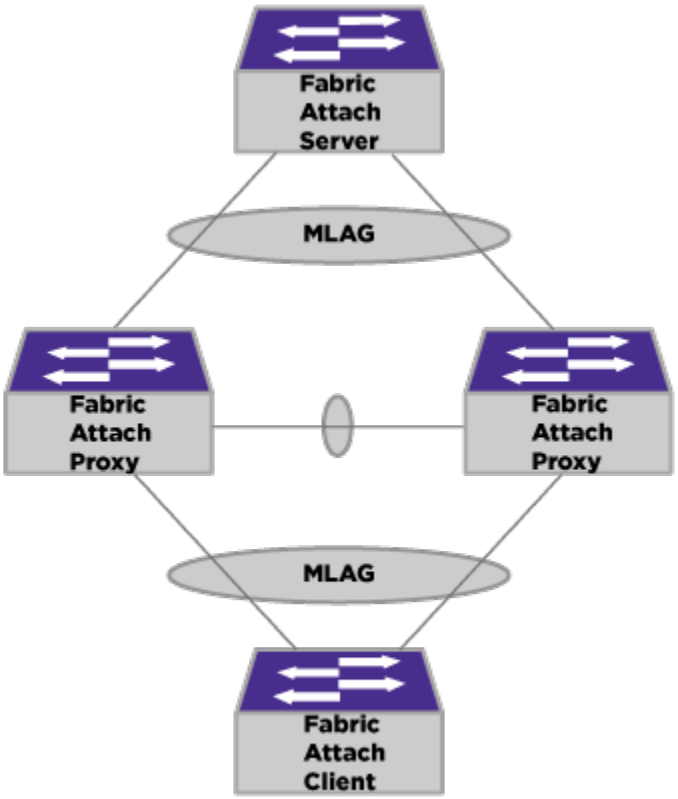


Figure 84: Fabric Attach with MLAG

Fabric Attach Proxy Support over MLAG

When an ExtremeXOS switch acts as a Fabric Attach proxy in an MLAG configuration, the VLAN/NSI mappings are sent on the master port of MLAG by the Fabric Attach client. The same mappings are sent on the slave ports of the MLAG by the Fabric Attach client in a delayed manner. This could result in data loss if the data packets are reaching the Fabric Attach proxy attached to the slave port of the MLAG. To address this, whenever VLAN/NSI mapping information is received on an MLAG port,

the Fabric Attach proxy forwards this VLAN/NSI mappings to the Fabric Attach server for authentication, and also checkpoints the information to the peer Fabric Attach proxy attached to the same MLAG. Respective VLANs received are created; ISC ports and MLAG ports are added to the respective VLAN. VLAN/NSI mappings are sent to the server for approval. After receiving VLAN/NSI mappings over the ISC link, the MLAG peer Fabric Attach proxy processes the metadata and with the received MLAG ID, it gets the local MLAG port and creates respective VLANs and ISCs. Additionally, the MLAG port is added to the respective VLANs. VLAN/NSI mappings are sent to the server for authentication. By checkpointing the NSI values across MLAG peer, redundancy is provided even if any of the client ports or server ports in MLAG go down.

For more information about MLAGs, see MLAG Overview on page 287.

Fabric Attach Server Support over MLAG

Fabric Attach server mode is also supported over MLAG, which is similar to proxy support on client-facing port side.

Fabric Attach Support Over W-MLAG

From ExtremeXOS 31.2, W-MLAG and single-homed MLAG support is provided for Fabric Attach (see Figure 85). Prior to ExtremeXOS 31.2, the mappings from MLAG clients were checkpointed to the proxy or server peer. But starting with ExtremeXOS 31.2, the mappings from non-MLAG clients are also checkpointed to the MLAG proxy or server peer node. If the proxy or server is part of a W-MLAG network, then the mapppings from all the clients, whether they are part of MLAG or not, are passed to all the peer nodes of the MLAG.

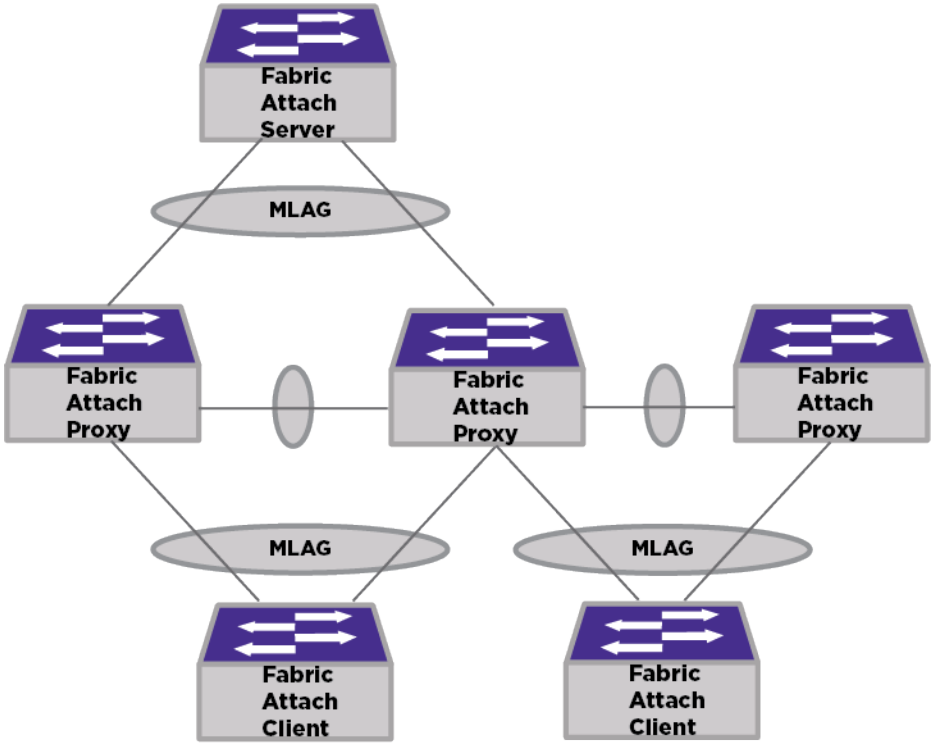


Figure 85: Fabric Attach with W-MLAG

Fabric Attach Authentication

By default, all ports are configured to the authentication disabled state. When enabled, the default key is used until otherwise configured. If the authentication fails, the Fabric Attach information is dropped whether or not authentication is enabled on the receiving port.

To set up Fabric Attach authentication, use the following command:

```
configure fabric attach ports [port_list | all] authentication [ disable
| enable | key {key | default | encrypted encrypted_key}]
```

You can set up authentication using either of the following methods:

- If you want to use the default key, enable authentication, and do not configure a key.
- To use a unique key (not the default), configure a key, and then enable authentication.

Be sure to configure authentication in the same manner on all applicable servers, proxies, and clients; and with MLAGs (Fabric Attach Support over Multi-switch Link Aggregation Groups (MLAGs) on page 793) on both peers.

RADIUS Vendor-Specific Attributes (VSAs) for Fabric Attach

To aid the proper authentication attributes sent from the RADIUS server to the device, several vendor-specific attributes (VSAs) exist to provide client-specific information from the device and information received from the Fabric Attach client at the time

it connects to the network. ExtremeXOS provides support in Fabric Attach for the following VSAs.

Note
Attributes are sent only for NetLogin MAC- and Dot1x-based authentications.

Table 44: VSA Information

VSA	Vendor Type	Description
Switch Mode (FA-Switch-Mode)	180	Defines the configuration of the FA device itself. It can take on 6 different values. • 0 or not sent—Switch has no concept of SPB/Fabric Attach. • 1—Switch is a FA server with SPB disabled and FA Enabled. • 2—Switch is a FA server with SPB enabled. • 3—Switch is a FA proxy connected to a FA Server with SPB disabled and FA enabled. • 4—Switch is a FA proxy connected to a FA Server with SPB enabled. • 5—Switch is a FA proxy standalone.
Client Identifer (FA-Client-ID)	181	Contains the MAC address of the FA Client received from the FA LLDP messaging.

Table 44: VSA Information (continued)

VSA	Vendor Type	Description
Client Type (FA-Client-Type)	182	Informs the RADIUS server of what the user requesting authentication's primary purpose is. This can allow the RADIUS server to return a consistent set of attributes for clients who have a similar purpose. Accepted values are in the range of 6–17 (client types, not server or proxy types) for this attribute, though values 1–17 are defined. • 1—FA Element Type Other • 2—FA Server • 3—FA Proxy • 4—FA-Server No Authentication • 5—FA Proxy No Authentication • 6—FA Client Wireless AP Type 1 [clients direct network attachment] (for example: AP 9100) • 7—FA Client Wireless AP Type 2 [clients tunneled to controller] • 8—FA Client FA Switch • 9—FA Client FA Router • 10—FA Client IP Phone • 11—FA Client IP Camera • 12 - FA Client IP Video • 13—FA Client Security Device (13) [FW, IPD/IDS etc.] • 14—FA Client Virtual Switch • 15—FA Client Server/Endpoint • 16—FA Client ONA SDN • 17—FA Client ONA SPBoIP
Pre-shared Key (PSK) Status (FA-Client-PSK)	183	Used to inform the RADIUS server of the Pre-shared Key status of the client being authenticated. If sent, it can take one of 5 binary values, 0,10,11,100,101. • Not Sent—Pre-share Key (PSK) used unknown. • 0—FA PSK not enabled on Fabric Attach Switch Port • 10—Failed. FA PSK used is default key • 11—Passed. FA PSK used is default key

Table 44: VSA Information (continued)

VSA	Vendor Type	Description
		• 100—Failed. FA PSK used is user-defined • 101—Passed. FA PSK used is user-defined

Zero Touch Client

Zero Touch Client (ZTC) adds addition functionality that enhances Fabric Attach (FA) by allowing static mappings for client types. Normally, Network Service Identifier (NSI) or Virtual LAN (VLAN) mappings are received from the Client through Link Layer Discovery Protocol (LLDP), or are configured statically on a port. Zero Touch Client provides a method to configure a mapping for a specific client type, so that all devices of that type that are discovered will have the same mapping. For example, all LLDP neighbors defined as Fabric Attach Client Phones can be configured to be added to VLAN "phoneVlan." In this scenario, all phones are treated the same without the Client requiring knowledge of this. Zero Touch Client NSI or VLAN mappings are added to the VLAN as the untagged VLAN for the port.

Zero Touch Client can configure only one mapping per client type since there can be only one untagged port VLAN. Zero Touch Client provides a method to configure port priority. If no priority is provided, the statically configured dot1p priority will be used.

Base Zero Touch (ZT) auto-attach functionality allows a device to acquire management VLAN information and use this data to facilitate device manageability and network configuration download across the network. When you have enabled Base ZT auto-attach operation, it extracts management VLAN data from the primary FA Server advertisements, and potentially uses this data to update the in-use management VLAN. This information can also be cascaded to FA Clients. An FA proxy can disable management VLAN cascading to clients device-wide, and disable by port.

Configuring Zero Touch Client

To configure Fabric Attach Zero Touch Client, run the following command:

```
configure fabric attach zero-touch-client client [vlan [vlan_name |
vlan_id] [nsi nsi | isid isid] {priority [priority | dot1p]} {enable |
disable} | none] | enable | disable]
```

To show configured Fabric Attach Zero Touch Clients, run the following command:

```
show fabric attach zero-touch-client
```

Fabric Attach Triggered Signaling

Version 32.3 adds support for Fabric Attach triggered signaling that replaces LLDP timer triggers to create VLAN/NSI binding. This feature provides faster learning of

VLAN/NSI bindings when a switch or link recovers from failure. Fabric Attach triggered signaling of VLAN/NSI mappings occurs when:

- A link between the client and proxy flaps
- A client switch reboots
- A proxy switch reboots
- A link between the server and proxy flaps
- A server switch reboots
- A new VLAN/NSI mapping is added in the client

Limitations

A proxy switch sends out the maximum of 94 VLAN/NSI mapping requests to a server based on the time the mappings are created. Only the first 94 mappings added in the VLAN/NSI DB are sent to the server for approval.

Configuring Fabric Attach

An ExtremeXOS Fabric Attach device automatically detects and exchanges information with attached Fabric Attach devices, except that a proxy does not exchange information with another proxy).

The following are optional user-configurable items:

- Static VLAN/NSI assignments
- Fabric Attach authentication on either a proxy or server
- Management VLAN (Fabric Attach server only)
- Uplink port (Fabric Attach proxy only)

Static VLAN/NSI Assignments

You explicitly create static Fabric Attach VLANs. Static VLANs can be mapped to a Network Service Identifier (NSI) or Individual Service Identifier (ISID).

To map a VLAN to an NSI/ISID, use the following command:

```
configure [{vlan} vlan_name |vlan vlan_id] add [nsi nsi | isid isid]
```

Mappings created with this command are persistent across reboots, whereas mappings by Fabric Attach clients and mappings requested by RADIUS/Policy/NetLogin are not persistent.

Note
While it is possible to configure an NSI mapping using the command `configure virtual-network vn_name vxlan vni [ vni | none]`, it is not recommended to do so when using Fabric Attach.

To unmap a VLAN from an NSI/ISID, use the following command:

```
configure [{vlan} vlan_name |vlan vlan_id] delete [nsi nsi | isid isid]
```

To display VLAN to NSI/ISID information, use the following command:

show fabric attach {{vlan} | {vlan} vlan_name | vlan vlan_id} assignments

If the VLAN/NSI mappings of type offset is present, then the offset NSI mappings are given the least precedence to be sent to the server for approval regardless of the time the mappings were created.

Static NSI Offset

Beginning with version 33.1.1, you can associate tagged VLANs to be mapped with a NSI value using static NSI offset configuration. When the NSI offset value is configured, any tagged VLAN that does not have a NSI associated will be applied with a NSI value, which is the sum of the configured NSI offset value and VLAN ID.

The NSI offset applied to a VLAN will be removed if an NSI of higher precedence is applied to the VLAN.

Use the following command to configure the NSI offset value:

configure fabric attach isid-nsi-offset isid_nsi_offset

Use the following command to view configured NSI offset information:

show fabric attach {{**vlan**} | {**vlan**} *vlan_name* | **vlan** *vlan_id*} **assignments**

Use the following command to unconfigure the NSI offset value:

configure fabric attach **isid-nsi-offset** *isid_nsi_offset* **none**

Fabric Attach Authentication

To set up authentication, use the following command:

configure fabric attach **ports** [*port_list* | **all**] **authentication** [**disable** | **enable** | **key** {*key* | **default** | **encrypted** *encrypted_key*}]

Note
When Fabric Attach authentication is configured on ports that are part of an , all ports on that MLAG must have the same Fabric Attach authentication configuration.

To view authentication information, use the following command:

show fabric attach **ports** [*port_list* | **all**] **authentication** {**detail**}

Standalone Proxy/Uplink Port

If there is no Fabric Attach-capable server available, configuring an uplink makes the device a standalone proxy. This works like a proxy, but without having discovered a Fabric Attach server. The uplink port is treated as the server port.

To enable standalone proxy mode and specify the uplink port, use the following command using the port keyword:

```
configure fabric attach uplink [port | none]
```

To disable standalone proxy mode, use the following command using the none keyword:

```
configure fabric attach uplink [port | none]
```

```
configure fabric attach ports [port_list | all] enable | disable]
```

Management VLAN

To specify the VLAN advertised to Fabric Attach clients for them to use as the management VLAN, use the following command:

```
configure fabric attach management-vlan [vlan_id | vlan_name | untagged
| none | forward [on | off] ]
```

```
configure fabric attach management-vlan ports [port_list | all] forward
[on | off]
```

The specified VLAN configuration on the Fabric Attach server is restricted to only tagged VLANs.

Zero Touch Server Management VLAN

Zero Touch Server enhances FA by allowing a management VLAN to be automatically propagated to FA proxy and FA clients. FA proxy and FA clients use the VLAN to obtain an IP address via Dynamic Host Configuration Protocol (DHCP) and connect to a management device.

Automatic Fallback Management VLAN Conditional Creation

The Fabric Attach feature automatically creates a fallback management VLAN (VLAN 4048) to ensure FA client ports have a management path during zero-touch provisioning. The switch creates this fallback VLAN only when zero-touch provisioning is running and the FA server is not sending a management VLAN.

Client ports are added to the fallback VLAN and the VLAN is advertised to FA clients via LLDP only after the FA server approves the Network Service Identifier (NSI) that the proxy requests. If the server rejects the NSI, the proxy automatically retries while zero-touch provisioning is active. Configuration changes are not applied during the retry period.

This conditional approach prevents disruption in environments where different management VLANs are already in use. The fallback VLAN is not created in deployments where zero-touch provisioning is inactive or when the FA server provides a management VLAN.

Viewing Fabric Attach Information

To display known fabric attach statistics information, use the following command:

```
show fabric attach statistics
```

Note
To clear these statistics, use the `clear counters` command.

To display known fabric attach elements (neighbors) information, use the following command:

```
show fabric attach elements
```

To display known fabric attach assignments (mappings) information, use the following command:

```
show fabric attach {{vlan} | {vlan} vlan_name | vlan vlan_id}
assignments
```

To display known fabric attach agent information, use the following command:

```
show fabric attach agent
```

To display known Fabric Attach authentication information, use the following command:

```
show fabric attach ports [port_list | all] authentication {detail}
```

Warnings for MLAG Ports

```
5.2.2.8-FASvr-DUT1-switch.110 # configure fabric attach ports 1 3 5 53 disable

Port 1:1 is a member port of 1:5. Configuring all members the same is recommended.
Port 1:3 is a member port of 1:5. Configuring all members the same is recommended.
Port 1:5 is a master port. Configuring all members the same is recommended.
Port 1:53 is a member port of 1:5. Configuring all members the same is recommended.
```

Fabric Attach Timeout Configurations

Version 33.2.1 and later supports the configuration of Fabric Attach agent/assignment timeout and Fabric Attach discovery timeout. To configure agent timeout, use the following command:

```
configure fabric attach agent-timeout timeout
```

To configure discovery timeout use the following command:

```
configure fabric attach discovery-timeout [timeout | none]
```

Supported Platforms

All platforms.

Fabric Attach and Extended Edge Switching Configuration Examples

ExtremeXOS Fabric Attach (FA) supports FA servers and FA Proxies to be Extended Edge Switching controlling bridge (CB) nodes connected with bridge port extenders (BPEs) in two different topologies:

- Single-arm ring—see Single-ARM MLAG Ring Topology Configuration Examples on page 803.
- Cross-connect MLAG—see Cross-Connect MLAG Topology Configuration Examples on page 815.

Limitations

Trusted ports on BPEs cannot be used to connect FA clients.

Single-ARM MLAG Ring Topology Configuration Examples

The following are three combinations through which you can connect FA clients to FA elements, such as FA proxies and FA servers:

- FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CBs FA Server Peers on page 803—FA server CB MLAG peers detect FA clients connected to BPE ports. CBs will add ISC ports and extended port to dynamic VLAN matching received VLAN/NSI mappings.
- FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CB FA Proxy Peers on page 808—FA server can be VSP or ExtremeXOS switches. BPEs and FA servers are attached by MLAGs.
- FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CB FA Standalone Peers—FA standalone proxy CB MLAG peers detect FA clients connected to BPE port. CBs will add ISC port and extended port to dynamic VLAN matching received VLAN/NSI mappings.

FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CBs FA Server Peers

This configuration example shows how to configure a Fabric Attach (FA) server Controlling Bridge (CB) to receive and approve VLAN/NSI mappings from an FA client connected to a bridge port extender (BPE).

In the following example:

- The FA proxy/client connected to BPE port sent VLAN/NSI mappings.
- The FA server CB MLAG peers see their LLDP neighbor as FA client.

- When mappings are received on BPE port, the FA server CB creates a dynamic VLAN and adds ISC ports and extended port to the dynamic VLAN matching received VLAN/NSI mappings.
- The FA server approves the mappings and moves the VLAN/NSI mapping to the active state in FA database.
- The FA server CB Extended Edge Switching ring peers act as VXLAN VTEPs.
- When creation of dynamic virtual networks is enabled (`configure virtual-network dynamic on`), Extended Edge Switching MLAG ring CB peers create dynamic virtual networks for VLAN-VNI pair that match the VLAN/NSI(I-SID) mappings in the FA database.

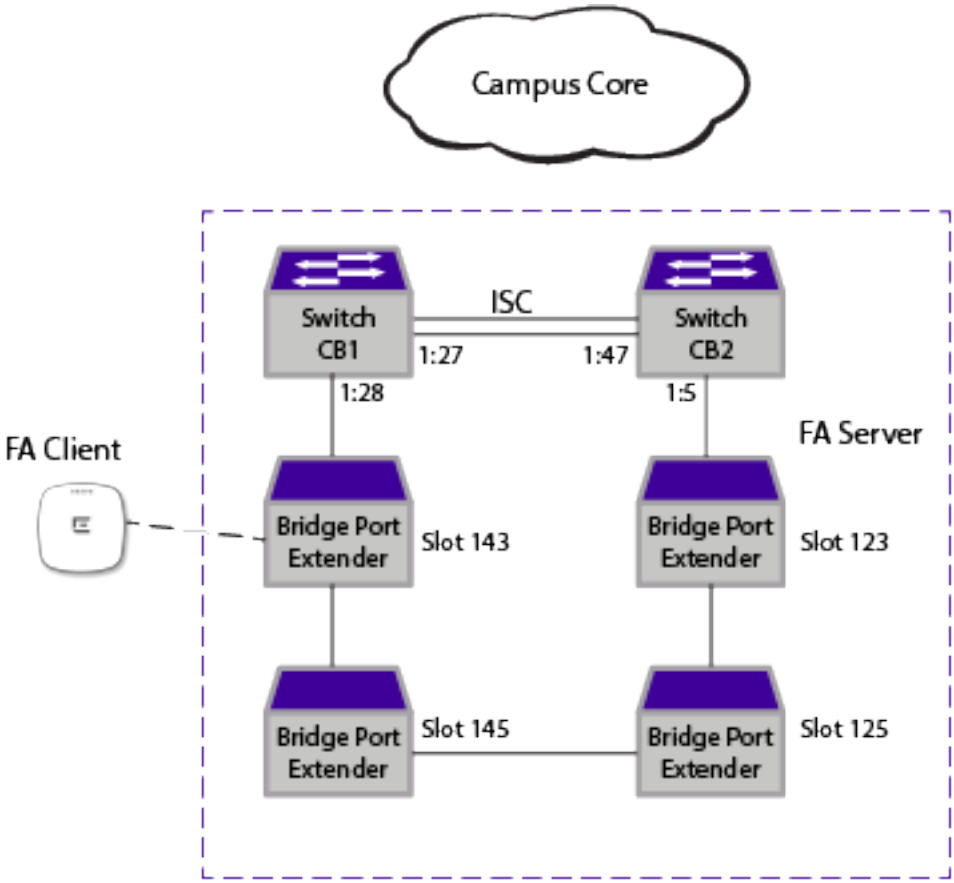


Figure 86: FA Clients Connect to BPE with Ring MLAG-Connected FA Server Peers

FA Client/Proxy Connected to BPE

Create VLAN/NSI mappings on the FA client switch connected to a BPE:

```
create vlan "VLAN_0505"
configure vlan VLAN_0505 tag 505
create vlan "VLAN_0707"
configure vlan VLAN_0707 tag 707
create vlan "VLAN_4004"
configure vlan VLAN_4004 tag 4004
configure vlan VLAN_0505 add nsi 5050
configure vlan VLAN_0707 add nsi 70707
configure vlan VLAN_4004 add nsi 41414
```

```
FAClient-450G2.5 # show fabric attach elements
Fabric Attach Mode: Proxy
                                                          Mgmt     Auto
System Id                      Port    Type               VLAN Tag Provision
------------------------------ ------- ------------------ ---- --- --------------
00-11-88-fe-f3-66-00-8f-00-06 17      Server (No Auth) 2002 Mix Disabled
* (CIT_30.4.0.524) FAClient-switch_model.6 #
 * (CIT_30.4.0.524) FAClient-switch_model.4 # show fabric attach assignments
Fabric Attach Mode: Proxy
Port     VLAN  VLAN Name                         Type     ISID/NSI  Status
-------  ----  -------------------------------  -------  --------  --------
          505  VLAN_0505                        Static       5050  Active
          707  VLAN_0707                        Static      70707  Active
         4004  VLAN_4004                        Static      41414  Active
```

```
FAClient-switch_model.5 # show lldp neighbors

        Neighbor           Neighbor                          Neighbor
Port    Chassis ID         Port ID            TTL     Age    System Name
================================================================================
17      00:11:88:FE:F3:66  143:6              120     1      CB1-465
================================================================================
NOTE: The Chassis ID and/or Port ID might be truncated to fit the screen.
```

Controlling Bridge 1 FA Server

```
Slot-1 CB1-465.57 # show fabric attach elements
Fabric Attach Mode: Server
                                                          Mgmt     Auto
System Id                      Port    Type               VLAN Tag Provision
------------------------------ ------- ------------------ ---- --- --------------
00-04-96-9a-8d-6c-00-01-00-0f 1:25    Proxy (No Auth)  2002 Mix Disabled
00-04-96-9c-d4-5a-00-01-00-12 1:26    Proxy (No Auth)  2002 Mix Disabled
00-04-96-9a-8d-45-00-01-00-2f 1:27    Server (No Auth) 2002 Mix Disabled
00-04-96-9a-8d-45-00-8f-00-19 1:28    Server (No Auth) 2002 Mix Disabled
00-04-96-98-a9-1a-00-01-00-11 143:6   Proxy (No Auth)  2002 Mix Disabled
```

```
Slot-1 CB1-465.59 # show fabric attach assignments
Fabric Attach Mode: Server
Port     VLAN  VLAN Name                         Type     ISID/NSI  Status
-------  ----  -------------------------------  -------  --------  --------
1:25      606  SYS_VLAN_0606                    Dynamic      6060  Active
1:25      756  SYS_VLAN_0756                    Dynamic     89002  Active
1:25      808  SYS_VLAN_0808                    Dynamic       808  Active
143:6     505  SYS_VLAN_0505                    Dynamic      5050  Active
143:6     707  SYS_VLAN_0707                    Dynamic     70707  Active
143:6    4004  SYS_VLAN_4004                    Dynamic     41414  Active
```

```
Slot-1 CB1-465.60 # show configuration vsm
#
# Module vsm configuration.
#
create mlag peer "Left-Peer-1"
configure mlag peer "Left-Peer-1" ipaddress 100.100.100.2 vr VR-Default
enable mlag port 1:25 peer "Left-Peer-1" id 200
enable mlag port 1:28 peer "Left-Peer-1" id 2001
(pacman debug) Slot-1 VPEX CB1-465.61 # show mlag peer
Multi-switch Link Aggregation Peers:

MLAG Peer         : Left-Peer-1
VLAN              : isc                    Virtual Router    : VR-Default
Local IP Address  : 100.100.100.1          Peer IP Address   : 100.100.100.2
MLAG ports        : 3                      Tx-Interval       : 1000 ms
```

```
Checkpoint Status : Up                      Peer Tx-Interval  : 1000 ms
Rx-Hellos         : 101274                  Tx-Hellos         : 101373
Rx-Checkpoint Msgs: 283537                  Tx-Checkpoint Msgs: 294709
Rx-Hello Errors   : 0                       Tx-Hello Errors   : 1408
Hello Timeouts    : 1                       Checkpoint Errors : 0
Up Time           : 1d:0h:28m:56s           Peer Conn.Failures: 1
Local MAC         : 00:11:88:fe:f3:66       Peer MAC          : 00:04:96:9a:8d:45
Config'd LACP MAC : None                    Current LACP MAC  : 00:04:96:9a:8d:45
Authentication    : None

Alternate path information: None
```

```
Slot-1 CB1-465.62 # show mlag ports

                Local                                                 Local   Remote
MLAG    Local   Link     Remote                              Peer     Fail    Fail
Id      Port    State    Link    Peer                        Status   Count   Count
================================================================================
200     1:25    A        Up      Left-Peer-1                 Up           0       0
2001    1:28    A        Virtual Left-Peer-1                 Up           0       1
================================================================================
Local Link State: A - Active, D - Disabled, R - Ready, NP - Port not present
Remote Link     : Up - One or more links are active on the remote switch,
                  Down - No links are active on the remote switch,
                  N/A - The peer has not communicated link state for this MLAG port,
                  Virtual - MLAG peer switch does not have physical port.

Number of Multi-switch Link Aggregation Groups  : 2
Convergence control                             : Conserve Access Lists
Reload Delay Interval                           : 30 seconds
Reload Delay                                    : Disabled
Link Up Isolation                               : Off
(pacman debug) Slot-1 VPEX CB1-465.63 #
* (pacman debug) Slot-1 VPEX CB1-465.13 # show virtual-network
Virtual Network                     Flags     Tenant VLAN
              Encap       ID     Encap Flags
================================================================================
SYS_VNET_00000808                   --S       SYS_VLAN_0808
              VXLAN      808     L-X
SYS_VNET_00005050                   --S       SYS_VLAN_0505
              VXLAN      5050    L-X
SYS_VNET_00006060                   --S       SYS_VLAN_0606
              VXLAN      6060    L-X
SYS_VNET_00041414                   --S       SYS_VLAN_4004
              VXLAN     41414    L-X
SYS_VNET_00070707                   --S       SYS_VLAN_0707
              VXLAN     70707    L-X
SYS_VNET_00089002                   --S       SYS_VLAN_0756
              VXLAN     89002    L-X
================================================================================
Flags: (S) System Configured, (T) OTM Configured, (V) OVSDB Configured
Encap Flags: (L) Local Endpoints Configured,
       (R) Remote Endpoints Associated,
       (X) Exclude Tag
----------------------------------------
Total number of Virtual Networks  : 6
Local Endpoints                   : 89.9.9.10 (VR-Default)
Network Ports [VXLAN]             : 1:1-32
Dynamic Virtual Networks          : On
```

Controlling Bridge 2 FA Server

```
Slot-1 CB2-670.35 # show fabric attach elements
Fabric Attach Mode: Server
```

```
                                                    Mgmt     Auto
System Id                      Port    Type             VLAN Tag Provision
------------------------------ ------- ---------------- ---- --- --------------
00-11-88-fe-f3-66-00-7b-00-19 1:5     Server (No Auth) 2002 Mix Disabled
00-04-96-9a-8d-6c-00-01-00-21 1:16    Proxy (No Auth)  2002 Mix Disabled
00-04-96-9a-8d-6c-00-01-00-0f 1:24    Proxy (No Auth)  2002 Mix Disabled
00-04-96-99-9e-31-00-01-00-33 1:33    Proxy (No Auth)  2002 Mix Disabled
00-11-88-fe-f3-66-00-01-00-1b 1:47    Server (No Auth) 2002 Mix Disabled
00-04-96-98-a9-1a-00-01-00-11 143:6   Proxy (No Auth)  2002 Mix Disabled
(pacman debug) Slot-1 VPEX CB2-670.36
#(pacman debug) Slot-1 VPEX CB2-670.36 # show fabric attach assignments
Fabric Attach Mode: Server
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
1:24      606  SYS_VLAN_0606                     Dynamic      6060  Active
1:24      756  SYS_VLAN_0756                     Dynamic     89002  Active
1:24      808  SYS_VLAN_0808                     Dynamic       808  Active
143:6     505  SYS_VLAN_0505                     Dynamic      5050  Active
143:6     707  SYS_VLAN_0707                     Dynamic     70707  Active
143:6    4004  SYS_VLAN_4004                     Dynamic     41414  Active
```

```
Slot-1 CB2-670.37 # show configuration vsm
#
# Module vsm configuration.
#
create mlag peer "Right-Peer-1"
configure mlag peer "Right-Peer-1" ipaddress 100.100.100.1 vr VR-Default
enable mlag port 1:5 peer "Right-Peer-1" id 1001
enable mlag port 1:24 peer "Right-Peer-1" id 200
(pacman debug) Slot-1 VPEX CB2-670.38 # show mlag peer
Multi-switch Link Aggregation Peers:

MLAG Peer         : Right-Peer-1
VLAN              : isc                   Virtual Router    : VR-Default
Local IP Address  : 100.100.100.2         Peer IP Address   : 100.100.100.1
MLAG ports        : 3                     Tx-Interval       : 1000 ms
Checkpoint Status : Up                    Peer Tx-Interval  : 1000 ms
Rx-Hellos         : 101309                Tx-Hellos         : 101408
Rx-Checkpoint Msgs: 294793                Tx-Checkpoint Msgs: 283640
Rx-Hello Errors   : 0                     Tx-Hello Errors   : 883
Hello Timeouts    : 1                     Checkpoint Errors : 0
Up Time           : 1d:0h:29m:29s         Peer Conn.Failures: 1
Local MAC         : 00:04:96:9a:8d:45     Peer MAC          : 00:11:88:fe:f3:66
Config'd LACP MAC : None                  Current LACP MAC  : 00:04:96:9a:8d:45
Authentication    : None

Alternate path information: None
```

```
Slot-1 CB2-670.39 # show mlag ports

               Local                                                 Local   Remote
MLAG    Local  Link     Remote                          Peer         Fail    Fail
Id      Port   State    Link    Peer                    Status       Count   Count
================================================================================
1001    1:5    A        Virtual Right-Peer-1            Up               0       1
200     1:24   A        Up      Right-Peer-1            Up               0       0
================================================================================
Local Link State: A - Active, D - Disabled, R - Ready, NP - Port not present
Remote Link     : Up - One or more links are active on the remote switch,
                  Down - No links are active on the remote switch,
                  N/A - The peer has not communicated link state for this MLAG port,
                  Virtual - MLAG peer switch does not have physical port.

Number of Multi-switch Link Aggregation Groups  : 2
```

```
Convergence control                              : Conserve Access Lists
Reload Delay Interval                            : 30 seconds
Reload Delay                                     : Disabled
Link Up Isolation                                : Off
```

```
Slot-1 CB2-670.15 # show virtual-network
Virtual Network                      Flags     Tenant VLAN
              Encap       ID     Encap Flags
================================================================================
SYS_VNET_00000808                    --S       SYS_VLAN_0808
              VXLAN      808     L-X
SYS_VNET_00005050                    --S       SYS_VLAN_0505
              VXLAN      5050    L-X
SYS_VNET_00006060                    --S       SYS_VLAN_0606
              VXLAN      6060    L-X
SYS_VNET_00041414                    --S       SYS_VLAN_4004
              VXLAN     41414    L-X
SYS_VNET_00070707                    --S       SYS_VLAN_0707
              VXLAN     70707    L-X
SYS_VNET_00089002                    --S       SYS_VLAN_0756
              VXLAN     89002    L-X
================================================================================
Flags: (S) System Configured, (T) OTM Configured, (V) OVSDB Configured
Encap Flags: (L) Local Endpoints Configured,
       (R) Remote Endpoints Associated,
       (X) Exclude Tag
---------------------------------------
Total number of Virtual Networks  : 6
Local Endpoints                   : 89.9.9.10 (VR-Default)
Network Ports [VXLAN]             : 1:1-64
Dynamic Virtual Networks          : On
```

FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CB FA Proxy Peers

This configuration example shows how to configure Fabric Attach (FA) proxy controlling bridge (CB) Extended Edge Switching Ring CB peers to receive and approve VLAN/NSI mappings from FA clients connected to BPEs.

In the following example:

- The FA proxy/client connected to BPE port sent VLAN/NSI mappings.
- The FA proxy CB MLAG peers see its LLDP neighbor as FA client.
- When mappings are received on a BPE port, the FA proxy CB creates a dynamic VLAN and adds ISC ports, uplink MLAG ports, and extended port to the dynamic VLAN that matches received VLAN/NSI mappings.
- The FA server approves the mappings and moves the VLAN/NSI mapping to the active state in FA database.
- The FA server MLAG peers act as VXLAN VTEPs.
- When creation of dynamic virtual networks is enabled (`configure virtual-network dynamic on`), Extended Edge Switching MLAG ring CB peers create dynamic virtual networks for VLAN-VNI pair that match the VLAN/NSI(I-SID) mappings in the FA database.

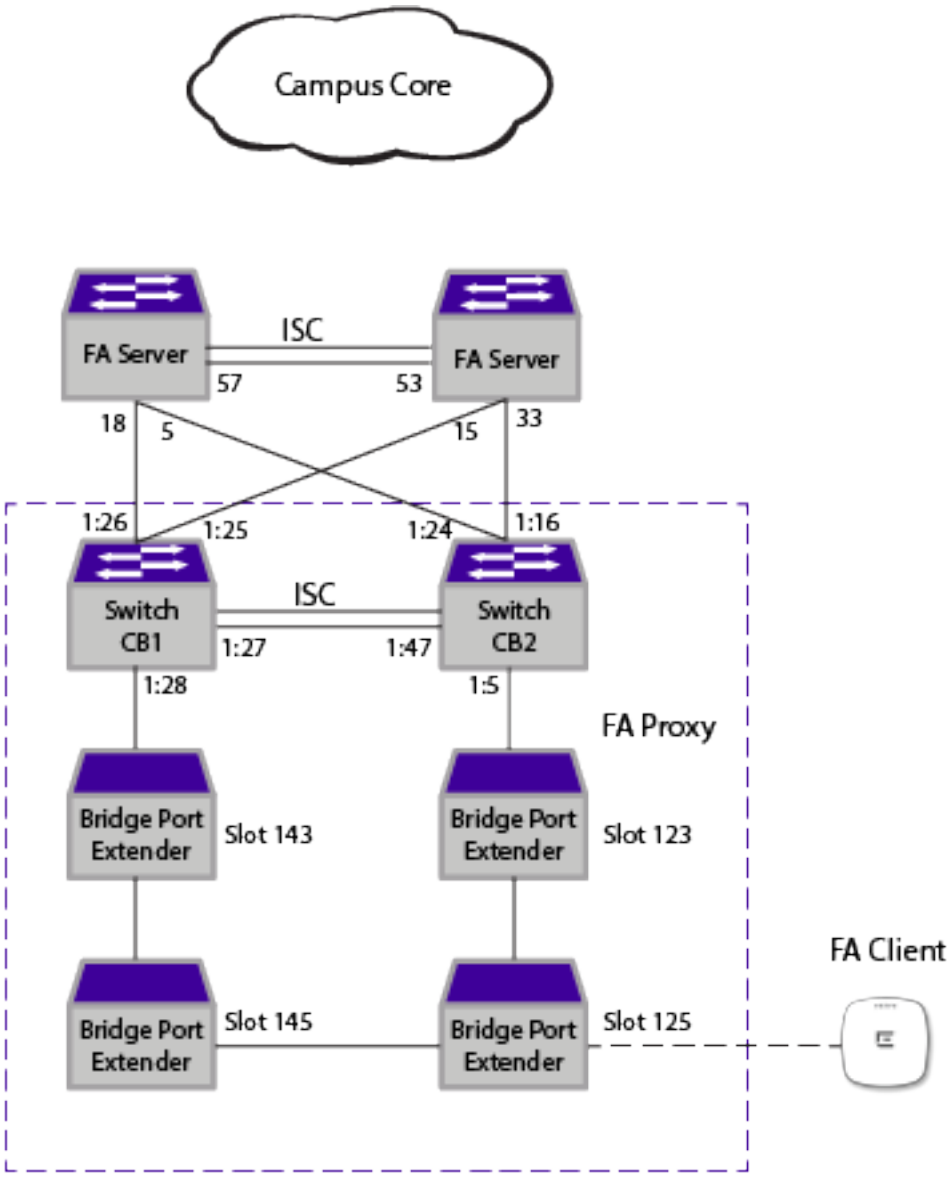


Figure 87: FA Clients Connect to BPE with Ring MLAG-Connected FA Proxy Peers

Controlling Bridge 1 FA Proxy

```
Slot-1 CB1-465.27 # show fabric attach assignments
Fabric Attach Mode: Proxy
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
143:6     505  SYS_VLAN_0505                     Dynamic      5050  Active
143:6     707  SYS_VLAN_0707                     Dynamic     70707  Active
143:6    4004  SYS_VLAN_4004                     Dynamic     41414  Active
```

```
Slot-1 CB1-465.27 # show fabric attach elements
Fabric Attach Mode: Proxy
                                                          Mgmt     Auto
System Id                      Port    Type               VLAN Tag Provision
------------------------------ ------- ----------------- ---- --- --------------
00-04-96-9c-d4-5a-00-01-00-12 1:26    Server (No Auth) 1000 Mix Disabled
00-04-96-9a-8d-45-00-01-00-2f 1:27    Proxy (No Auth)  1000 Mix Disabled
```

```
00-04-96-9a-8d-45-00-8f-00-19 1:28    Proxy (No Auth)  1000 Mix Disabled
00-04-96-98-a9-1a-00-01-00-11 143:6   Switch           1000 Mix Disabled
* (pacman debug) Slot-1 VPEX CB1-465.28 #
* (pacman debug) Slot-1 VPEX CB1-465.28 # show fabric attach assignments
Fabric Attach Mode: Proxy
Port     VLAN  VLAN Name                         Type     ISID/NSI  Status
-------  ----  -------------------------------  -------  --------  --------
143:6     505  SYS_VLAN_0505                    Dynamic      5050  Active
143:6     707  SYS_VLAN_0707                    Dynamic     70707  Active
143:6    4004  SYS_VLAN_4004                    Dynamic     41414  Active
```

```
Slot-1 CB1-465.29 # show lldp ports 143:6 neighbors detailed

--------------------------------------------------------------------------------
LLDP Port 143:6 detected 1 neighbor
  Neighbor: 00:04:96:98:A9:1A/17, age 20 seconds
    - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:98:A9:1A
    - Port ID type: ifName (5)
      Port ID     : "17"
    - Time To Live: 120 seconds
    - System Name: "FAClient-switch_model"
    - System Description: "ExtremeXOS (switch_model) version 30.4.0.524 30.\
                           4.0.524 by release-manager on Fri Oct 25 15:01:31\
                            EDT 2019"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 8
      State          : 0
      Management Vlan: 1000
      SystemId       : 00:04:96:98:a9:1a
      Link Info      : 00-01-00-11

    - Extreme Fabric Attach NSI(I-SID)/VLAN map
    Status  VlanID  NSI(I-SID)
    ------  ------  --------
         1     505      5050
         1     707     70707
         1    4004     41414
```

Controlling Bridge 2 FA Proxy

```
Slot-1 CB2-670.25 # show fabric attach assignments
Fabric Attach Mode: Proxy
Port     VLAN  VLAN Name                         Type     ISID/NSI  Status
-------  ----  -------------------------------  -------  --------  --------
143:6     505  SYS_VLAN_0505                    Dynamic      5050  Active
143:6     707  SYS_VLAN_0707                    Dynamic     70707  Active
143:6    4004  SYS_VLAN_4004                    Dynamic     41414  Active
```

```
Slot-1 CB2-670.25 # show fabric attach elements
Fabric Attach Mode: Proxy
                                                      Mgmt     Auto
System Id                     Port    Type            VLAN Tag Provision
----------------------------- ------- --------------- ---- --- --------------
00-11-88-fe-f3-66-00-7b-00-19 1:5     Proxy (No Auth)  1000 Mix Disabled
00-04-96-9c-d4-5a-00-01-00-05 1:24    Server (No Auth) 1000 Mix Disabled
00-04-96-99-9e-31-00-01-00-33 1:33    Switch           1000 Mix Disabled
00-11-88-fe-f3-66-00-01-00-1b 1:47    Proxy (No Auth)  1000 Mix Disabled
00-04-96-98-a9-1a-00-01-00-11 143:6   Switch           1000 Mix Disabled
```

```
Slot-1 CB2-670.26 # show lldp ports 143:6 neighbors detailed

--------------------------------------------------------------------------------
```

```
LLDP Port 143:6 detected 1 neighbor
  Neighbor: 00:04:96:98:A9:1A/17, age 22 seconds
    - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:98:A9:1A
    - Port ID type: ifName (5)
      Port ID      : "17"
    - Time To Live: 120 seconds
    - System Name: "FAClient-switch_model"
    - System Description: "ExtremeXOS (switch_model) version 30.4.0.524 30.\
                           4.0.524 by release-manager on Fri Oct 25 15:01:31\
                            EDT 2019"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 8
      State          : 0
      Management Vlan: 1000
      SystemId       : 00:04:96:98:a9:1a
      Link Info      : 00-01-00-11

    - Extreme Fabric Attach NSI(I-SID)/VLAN map
    Status  VlanID  NSI(I-SID)
    ------  ------  --------
         2     505      5050
         2     707     70707
         2    4004     41414
```

FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CB FA Standalone Peers

Fabric Attach (FA) standalone proxy introduces FA proxy functionality in environments without an FA server. FA provisions VLANs on the FA standalone proxy switch for itself, and for connected FA clients, and then adds the VLAN membership to the uplink port facing the core (non-FA) switch or router for connection to existing services. The provisioned VLANs at the access layer switch *must* match the VLANs on the opposite end of the link at the core/aggregation layer switch for communications to operate. You can use this design with an existing legacy core network infrastructure (any vendor).

This configuration example shows how to configure FA standalone CB Extended Edge Switching MLAG peers over a ring to receive and approve VLAN/NSI mappings from an FA client connected to BPEs.

In the following example:

- FA client connected to BPE port sent VLAN/NSI mappings.
- The FA standalone proxy CB MLAG peers see their LLDP neighbor as FA client.
- When mappings are received on BPE ports, the FA standalone proxy CB creates a dynamic VLAN, and adds ISC port and extended port to the dynamic VLAN that matches received VLAN/NSI mappings.
- The command `show fabric attach` **`statistics`** using the assignments option shows ISID/NSI as N/A on Extended Edge Switching standalone CBs.

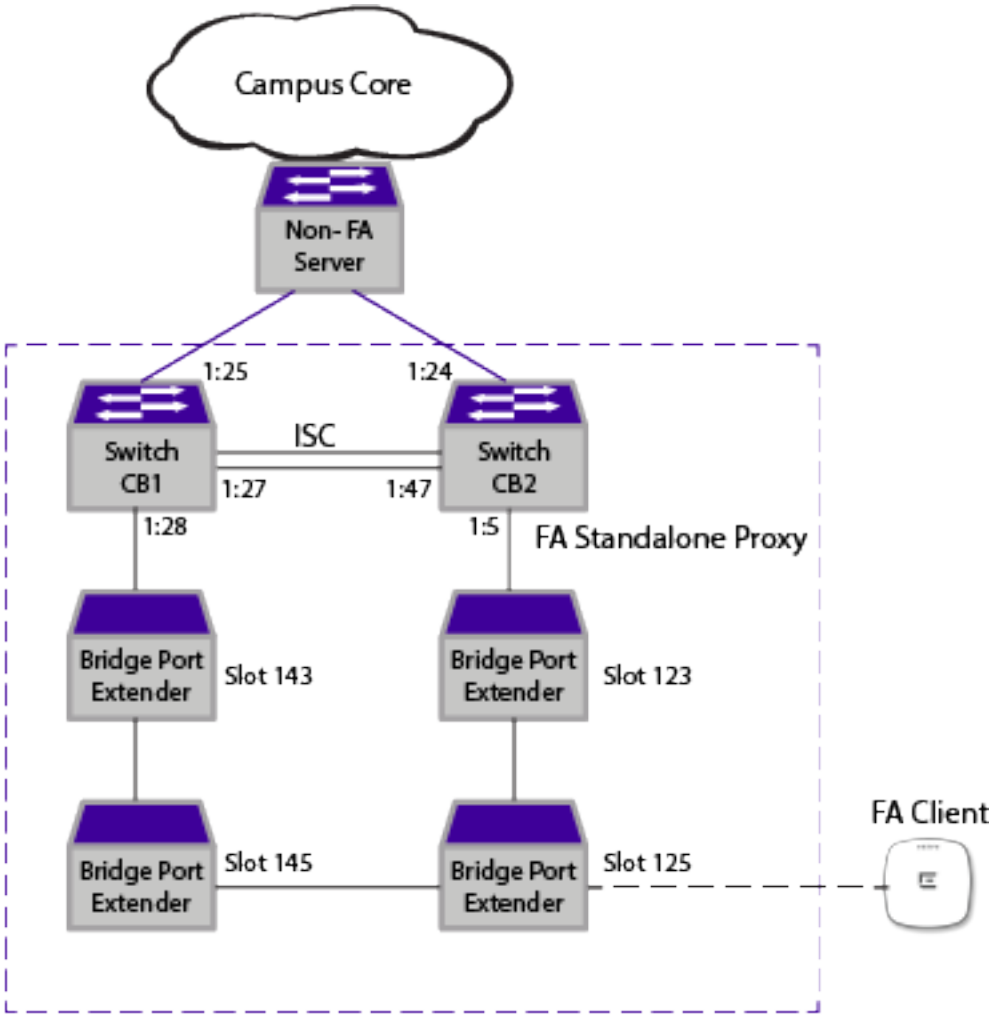


Figure 88: FA Clients Connected to BPEs with Extended Edge Switching Ring MLAG-Connected CBs FA Standalone Peers

Controlling Bridge1 FA Standalone Proxy

```
Slot-1 CB1-465.11 # show configuration lldp
#
# Module lldp configuration.
#
configure fabric attach uplink 1:26
```

```
Slot-1 CB1-465.12 # show fabric attach assignments
Fabric Attach Mode: Standalone Proxy
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
143:6     505  SYS_VLAN_0505                      Dynamic  N/A       Active
143:6     707  SYS_VLAN_0707                      Dynamic  N/A       Active
143:6    4004  SYS_VLAN_4004                      Dynamic  N/A       Active
* (pacman debug) Slot-1 VPEX CB1-465.13 # show lldp ports 143:6 neighbors detailed

--------------------------------------------------------------------------------
LLDP Port 143:6 detected 1 neighbor
  Neighbor: 00:04:96:98:A9:1A/17, age 19 seconds
    - Chassis ID type: MAC address (4)
```

```
      Chassis ID     : 00:04:96:98:A9:1A
    - Port ID type: ifName (5)
      Port ID      : "17"
    - Time To Live: 120 seconds
    - System Name: "FAClient-switch_model"
    - System Description: "ExtremeXOS (switch_model) version 30.4.0.524 30.\
                           4.0.524 by release-manager on Fri Oct 25 15:01:31\
                            EDT 2019"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 8
      State          : 0
      Management Vlan: 0
      SystemId       : 00:04:96:98:a9:1a
      Link Info      : 00-01-00-11

    - Extreme Fabric Attach NSI(I-SID)/VLAN map
    Status  VlanID  NSI(I-SID)
    ------  ------  --------
         2     505      5050
         2     707     70707
         2    4004     41414
```

```
Slot-1 CB1-465.14 # show vlan
Untagged ports auto-move: Inform
-------------------------------------------------------------------------------------------
-----
Name            VID  Protocol Addr       Flags                        Proto  Ports
Virtual
                                                                             Active
router
                                                                             /Total
-------------------------------------------------------------------------------------------
-----
Default         1    --------------------------------T---------------  ANY    5 /151 VR-
Default
isc             1001 100.100.100.1  /24  ------I---------------------  ANY    1 /1   VR-
Default
Mgmt            4095 10.127.16.12   /24  ----------------------------  ANY    1 /1   VR-
Mgmt
SYS_VLAN_0505   505  ---------------------------------------d--------  ANY    3 /3   VR-
Default
SYS_VLAN_0707   707  ---------------------------------------d--------  ANY    3 /3   VR-
Default
SYS_VLAN_1000   1000 ---------------------------------------d--------  ANY    3 /3   VR-
Default
SYS_VLAN_4004   4004 ---------------------------------------d--------  ANY    3 /3   VR-
Default
-------------------------------------------------------------------------------------------
-----
```

```
Slot-1 CB1-465.15 # show vlan 505 | include Tag:
VLAN 505: show vlan SYS_VLAN_0505 | include Tag:
       Tag:   *1:26GF,*143:6F,*1:27gF
```

Controlling Bridge 2 FA Standalone Proxy

```
Slot-1 VPEX CB2-670.9 # show configuration lldp
#
# Module lldp configuration.
#
configure fabric attach uplink 1:24
```

```
Slot-1 CB2-670.10 # show fabric attach assignments
Fabric Attach Mode: Standalone Proxy
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  ------------------------------  -------  --------  --------
143:6     505  SYS_VLAN_0505                     Dynamic  N/A       Active
143:6     707  SYS_VLAN_0707                     Dynamic  N/A       Active
143:6    4004  SYS_VLAN_4004                     Dynamic  N/A       Active
* (pacman debug) Slot-1 VPEX CB2-670.11 #
* (pacman debug) Slot-1 VPEX CB2-670.11 # show lldp ports 143:6 neighbors detailed

--------------------------------------------------------------------------------
LLDP Port 143:6 detected 1 neighbor
  Neighbor: 00:04:96:98:A9:1A/17, age 18 seconds
    - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:98:A9:1A
    - Port ID type: ifName (5)
      Port ID     : "17"
    - Time To Live: 120 seconds
    - System Name: "FAClient-switch_model"
    - System Description: "ExtremeXOS (switch_model) version 30.4.0.524 30.\
                           4.0.524 by release-manager on Fri Oct 25 15:01:31\
                            EDT 2019"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 8
      State          : 0
      Management Vlan: 0
      SystemId       : 00:04:96:98:a9:1a
      Link Info      : 00-01-00-11

    - Extreme Fabric Attach NSI(I-SID)/VLAN map
    Status  VlanID  NSI(I-SID)
    ------  ------  --------
         2     505      5050
         2     707     70707
         2    4004     41414
```

```
Slot-1 CB2-670.12 # show vlan
Untagged ports auto-move: Inform
--------------------------------------------------------------------------------------------------
-----
Name            VID  Protocol Addr       Flags                        Proto  Ports
Virtual
                                                                             Active
router
                                                                             /Total
--------------------------------------------------------------------------------------------------
-----
Default         1    --------------------------------T---------------  ANY    6 /183 VR-
Default
isc             1001 100.100.100.2  /24  ------I---------------------  ANY    1 /1   VR-
Default
Mgmt            4095 10.127.16.6    /24  ---------------------------  ANY    1 /1   VR-
Mgmt
SYS_VLAN_0505   505  ---------------------------------------d--------  ANY    3 /3   VR-
Default
SYS_VLAN_0707   707  ---------------------------------------d--------  ANY    3 /3   VR-
Default
SYS_VLAN_1000   1000 ---------------------------------------d--------  ANY    4 /4   VR-
Default
SYS_VLAN_4004   4004 ---------------------------------------d--------  ANY    3 /3   VR-
Default
--------------------------------------------------------------------------------------------------
-----
```

```
Slot-1 CB2-670.13 # show vlan 505 | include Tag:
VLAN 505: show vlan SYS_VLAN_0505 | include Tag:
       Tag:   *1:24GF,*143:6F,*1:47gF
```

Cross-Connect MLAG Topology Configuration Examples

Cross-Connect MLAG Topology Configuration Example

In the following example:

- The Fabric Attach (FA) client connected to bridge port extender (BPE) port sends VLAN/NSI mappings to the FA proxy controlling bridge (CB) peers connected in cross-connect topology.
- The FA proxy CB MLAG peers see their LLDP neighbor as the FA client.
- When mappings are received on a BPE port, the FA proxy CB creates a dynamic VLAN, and adds ISC port, extended port, and uplink server port to the dynamic VLAN that matches the received VLAN/NSI mappings.
- Uplink ports connected to the FA server MLAG peers on the CB MLAG peers should be MLAG enabled.
- FA server peers in following example are non-Extended Edge Switching nodes, which can be ExtremeXOS or VSP switches.
- The FA server approves the mappings and moves the VLAN/NSI mapping to the active state in FA database. The ExtremeXOS FA server MLAG peers act as VXLAN VTEPs.
- When creation of dynamic virtual networks is enabled (`configure virtual-network dynamic on`), MLAG peers create dynamic virtual networks for the VLAN-VNI pair that matches the VLAN/NSI(I-SID) mappings in the FA database.

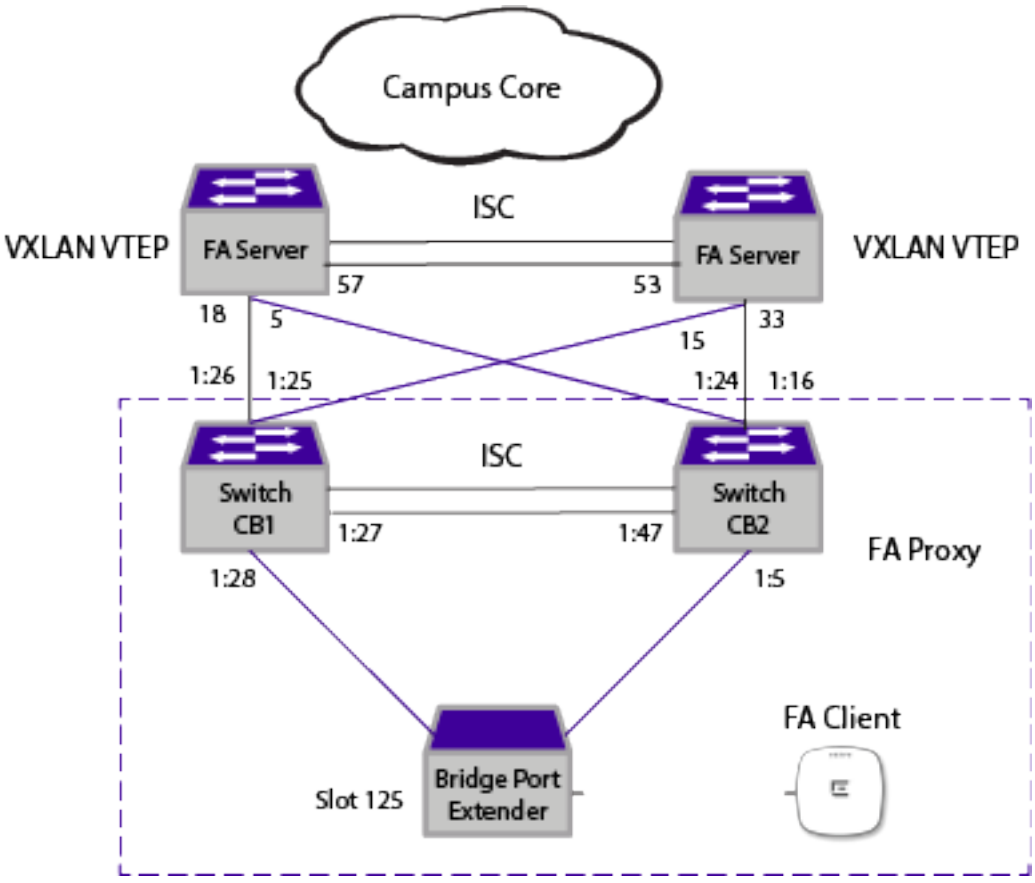


Figure 89: Cross-Connect MLAG Topology

Controlling Bridge 1 (FA Proxy CB1)

```
# show lldp ports 125:9 neighbors detailed

--------------------------------------------------------------------------------
LLDP Port 125:9 detected 1 neighbor
  Neighbor: 00:04:96:9D:42:32/15, age 3 seconds
   - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:9D:42:32
    - Port ID type: ifName (5)
      Port ID      : "15"
    - Time To Live: 120 seconds
    - System Name: "5720"
    - System Description: "Switch Engine switch version 32.4.1 by release-manager on Mon
Nov 4 22:39:20\
                          EST 2023"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 5
      State          : 0
      Management Vlan: 0
      SystemId       : 00:04:96:9d:42:32
      Link Info      : 00-01-00-0f
```

```
# show fabric attach assignments
Fabric Attach Mode: Proxy
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
125:5     100  SYS_VLAN_0100                     Dynamic   2800100  Active
```

```
125:5     101  SYS_VLAN_0101                     Dynamic   2800101  Active
(pacman debug) Slot-1 VPEX CB1-465.12 # show fabric attach elements
Fabric Attach Mode: Proxy
                                                          Mgmt     Auto
System Id                      Port    Type               VLAN Tag Provision
------------------------------ ------- ---------------- ---- --- --------------
00-04-96-9d-42-32-00-01-00-1f 1:1     Proxy (No Auth)  None Mix Disabled
00-04-96-9a-8d-6c-00-01-00-0f 1:25    Server (No Auth) None Mix Disabled
00-04-96-9c-d4-5a-00-01-00-12 1:26    Server (No Auth) None Mix Disabled
00-04-96-9a-8d-45-00-01-00-2f 1:27    Proxy (No Auth)  None Mix Disabled
00-04-96-98-a9-1a-00-01-00-11 125:5   Switch           None Mix Disabled
00-04-96-9d-42-32-00-01-00-0f 125:9   Proxy (No Auth)  None Mix Disabled
```

```
# show vlan 100 | include Tag
VLAN 100: show vlan SYS_VLAN_0100 | include Tag
    Admin State:         Enabled     Tagging:   802.1Q Tag 100
       Tag:   *125:5F,*1:27gF,*1:25gGF
                  (x) VMAN Tag Translated port
(pacman debug) Slot-1 VPEX CB1-465.14 #
(pacman debug) Slot-1 VPEX CB1-465.26 # show mlag ports

                Local                                                  Local   Remote
MLAG    Local   Link     Remote                            Peer      Fail    Fail
Id      Port    State    Link    Peer                      Status    Count   Count
================================================================================
20      1:25    A        Up      right-peer                Up          0       0
10      1:28    A        Up      right-peer                Up          0       0
================================================================================
Local Link State: A - Active, D - Disabled, R - Ready, NP - Port not present
Remote Link     : Up - One or more links are active on the remote switch,
                  Down - No links are active on the remote switch,
                  N/A - The peer has not communicated link state for this MLAG port,
                  Virtual - MLAG peer switch does not have physical port.

Number of Multi-switch Link Aggregation Groups  : 2
Convergence control                             : Conserve Access Lists
Reload Delay Interval                           : 30 seconds
Reload Delay                                    : Disabled
Link Up Isolation                               : Off
```

```
27 # show mlag peer
Multi-switch Link Aggregation Peers:

MLAG Peer         : right-peer
VLAN              : isc                    Virtual Router    : VR-Default
Local IP Address  : 45.0.0.1               Peer IP Address   : 45.0.0.2
MLAG ports        : 2                      Tx-Interval       : 1000 ms
Checkpoint Status : Up                     Peer Tx-Interval  : 1000 ms
Rx-Hellos         : 18771                  Tx-Hellos         : 18772
Rx-Checkpoint Msgs: 14572                  Tx-Checkpoint Msgs: 17087
Rx-Hello Errors   : 0                      Tx-Hello Errors   : 826
Hello Timeouts    : 1                      Checkpoint Errors : 0
Up Time           : 0d:5h:4m:33s           Peer Conn.Failures: 1
Local MAC         : 00:11:88:fe:f3:66      Peer MAC          : 00:04:96:9a:8d:45
Config'd LACP MAC : None                   Current LACP MAC  : 00:04:96:9a:8d:45
Authentication    : None

Alternate path information: None
```

```
# show sharing
Load Sharing Monitor
Config    Current Agg     Min    Ld Share        Ld Share  Agg Link  Link Up
Master    Master  Control Active Algorithm Flags Group     Mbr State Transitions
```

```
================================================================================
  1:25   1:25     LACP       1    L2        A     1:25      Y     A       0
                                  L2              1:26      Y     A       0
  1:27   1:27     LACP       1    L2        A     1:27      Y     A       0
  1:28   1:28     LACP       1    L2        A     1:28      Y     A       0
================================================================================
Link State: A-Active, D-Disabled, R-Ready, NP-Port not present, L-Loopback
Minimum Active: (<) Group is down. # active links less than configured minimum
Load Sharing Algorithm: (L2) Layer 2 address based, (L3) Layer 3 address based
                        (L3_L4) Layer 3 address and Layer 4 port based
                        (custom) User-selected address-based configuration
Custom Algorithm Configuration: ipv4 L3-and-L4, xor
Custom Hash Seed: Switch MAC address (0x88FEF366)
Flags:
    A - All: Distribute to all members,
    d - Dynamically created shared port,
    L - Local Slot: Distribute to members local to ingress slot,
    P - Port Lists: Distribute to per-slot configurable subset of members,
    R - Resilient Hashing enabled.
Number of load sharing trunks: 3
```

```
# show configuration vsm
#
# Module vsm configuration.
#
create mlag peer "right-peer"
configure mlag peer "right-peer" ipaddress 45.0.0.2 vr VR-Default
enable mlag port 1:25 peer "right-peer" id 20
enable mlag port 1:28 peer "right-peer" id 10
* (pacman debug) Slot-1 VPEX CB1-465.22 # show configuration vpex
#
# Module vpex configuration.
#
configure vpex port 1:28 slot 125
configure vpex ring rebalancing off
* (pacman debug) Slot-1 VPEX CB1-465.23 #
* (pacman debug) Slot-1 VPEX CB1-465.23 # show vpex topology

Topology: Cascade
  Native cascade port 1:28
    Slots: 125
```

Controlling Bridge1 (FA Proxy CB2)

```
# show fabric attach elements
Fabric Attach Mode: Proxy
                                                        Mgmt     Auto
System Id                      Port    Type             VLAN Tag Provision
------------------------------ ------- ---------------- ---- --- --------------
00-04-96-9a-8d-6c-00-01-00-21 1:16    Server (No Auth) None Mix Disabled
00-04-96-9c-d4-5a-00-01-00-05 1:24    Server (No Auth) None Mix Disabled
00-04-96-99-9e-31-00-01-00-33 1:33    Switch           None Mix Disabled
00-11-88-fe-f3-66-00-01-00-1b 1:47    Proxy (No Auth)  None Mix Disabled
00-04-96-98-a9-1a-00-01-00-11 125:5   Switch           None Mix Disabled
00-04-96-9d-42-32-00-01-00-0f 125:9   Proxy (No Auth)  None Mix Disabled
(pacman debug) Slot-1 VPEX CB2-670.2 # show fabric attach assignments
Fabric Attach Mode: Proxy
Port     VLAN  VLAN Name                         Type     ISID/NSI  Status
-------  ----  -------------------------------  -------  --------  --------
125:5     100  SYS_VLAN_0100                    Dynamic   2800100  Active
125:5     101  SYS_VLAN_0101                    Dynamic   2800101  Active
(pacman debug) Slot-1 VPEX CB2-670.3 #
(pacman debug) Slot-1 VPEX CB2-670.3 # show lldp ports 125:5 neighbors detailed
```

```
--------------------------------------------------------------------------------
LLDP Port 125:5 detected 1 neighbor
  Neighbor: 00:04:96:98:A9:1A/17, age 18 seconds
    - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:98:A9:1A
    - Port ID type: ifName (5)
      Port ID      : "17"
    - Time To Live: 120 seconds
    - System Name: "FAClient-450G2"
    - System Description: "ExtremeXOS (X450G2-24t-G4) version 30.4.0.524 30.\
                           4.0.524 by release-manager on Fri Oct 25 15:01:31\
                            EDT 2019"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 8
      State          : 0
      Management Vlan: 0
      SystemId       : 00:04:96:98:a9:1a
      Link Info      : 00-01-00-11

    - Extreme Fabric Attach NSI(I-SID)/VLAN map
    Status  VlanID  NSI(I-SID)
    ------  ------  --------
         2     100    2800100
         2     101    2800101
```

```
# show vlan 100 | include Tag
VLAN 100: show vlan SYS_VLAN_0100 | include Tag
    Admin State:         Enabled     Tagging:   802.1Q Tag 100
       Tag:   *125:5F,*1:47gF,*1:16gGF
                  (x) VMAN Tag Translated port
```

```
# show mlag ports

                Local                                                 Local   Remote
MLAG    Local   Link     Remote                          Peer         Fail    Fail
Id      Port    State    Link    Peer                    Status       Count   Count
================================================================================
20      1:16    A        Up      left-peer               Up           0       0
10      1:29    A        Up      left-peer               Up           0       0
================================================================================
Local Link State: A - Active, D - Disabled, R - Ready, NP - Port not present
Remote Link     : Up - One or more links are active on the remote switch,
                  Down - No links are active on the remote switch,
                  N/A - The peer has not communicated link state for this MLAG port,
                  Virtual - MLAG peer switch does not have physical port.

Number of Multi-switch Link Aggregation Groups  : 2
Convergence control                             : Conserve Access Lists
Reload Delay Interval                           : 30 seconds
Reload Delay                                    : Disabled
Link Up Isolation                               : Off
```

```
# show mlag peer
Multi-switch Link Aggregation Peers:

MLAG Peer         : left-peer
VLAN              : isc                    Virtual Router    : VR-Default
Local IP Address  : 45.0.0.2               Peer IP Address   : 45.0.0.1
MLAG ports        : 2                      Tx-Interval       : 1000 ms
Checkpoint Status : Up                     Peer Tx-Interval  : 1000 ms
Rx-Hellos         : 18811                  Tx-Hellos         : 18838
Rx-Checkpoint Msgs: 17113                  Tx-Checkpoint Msgs: 14591
Rx-Hello Errors   : 0                      Tx-Hello Errors   : 103
```

```
Hello Timeouts    : 1                       Checkpoint Errors : 0
Up Time           : 0d:5h:5m:12s            Peer Conn.Failures: 1
Local MAC         : 00:04:96:9a:8d:45       Peer MAC          : 00:11:88:fe:f3:66
Config'd LACP MAC : None                    Current LACP MAC  : 00:04:96:9a:8d:45
Authentication    : None

Alternate path information: None
```

```
# show sharing
Load Sharing Monitor
Config    Current Agg     Min    Ld Share        Ld Share  Agg Link  Link Up
Master    Master  Control Active Algorithm Flags Group     Mbr State Transitions
================================================================================
  1:16   1:16     LACP       1    L2        A     1:16      Y     A       0
                                  L2              1:24      Y     A       0
  1:29   1:29     LACP       1    L2        A     1:29      Y     A       0
  1:47   1:47     LACP       1    L2        A     1:47      Y     A       0
================================================================================
Link State: A-Active, D-Disabled, R-Ready, NP-Port not present, L-Loopback
Minimum Active: (<) Group is down. # active links less than configured minimum
Load Sharing Algorithm: (L2) Layer 2 address based, (L3) Layer 3 address based
                        (L3_L4) Layer 3 address and Layer 4 port based
                        (custom) User-selected address-based configuration
Custom Algorithm Configuration: ipv4 L3-and-L4, xor
Custom Hash Seed: Switch MAC address (0x969A8D45)
Flags:
    A - All: Distribute to all members,
    d - Dynamically created shared port,
    L - Local Slot: Distribute to members local to ingress slot,
    P - Port Lists: Distribute to per-slot configurable subset of members,
    R - Resilient Hashing enabled.
Number of load sharing trunks: 3
```

```
# show configuration vsm
#
# Module vsm configuration.
#
create mlag peer "left-peer"
configure mlag peer "left-peer" ipaddress 45.0.0.1 vr VR-Default
enable mlag port 1:16 peer "left-peer" id 20
enable mlag port 1:29 peer "left-peer" id 10
```

```
14 # show configuration vpex
#
# Module vpex configuration.
#
configure vpex port 1:29 slot 125
configure vpex ring rebalancing off
* (pacman debug) Slot-1 VPEX CB2-670.15 #
```

```
# show vpex topology

Topology: Cascade
  Native cascade port 1:29
    Slots: 125
```

FA Server 1

```
# show virtual-network
Virtual Network                      Flags     Tenant VLAN
              Encap       ID     Encap Flags
================================================================================
SYS_VNET_02800100                    --S       SYS_VLAN_0100
```

```
              VXLAN    2800100   LRX
SYS_VNET_02800101                   --S       SYS_VLAN_0101
              VXLAN    2800101   LRX
================================================================================
Flags: (S) System Configured, (T) OTM Configured, (V) OVSDB Configured
Encap Flags: (L) Local Endpoints Configured,
       (R) Remote Endpoints Associated,
       (X) Exclude Tag
----------------------------------------
Total number of Virtual Networks  : 2
Local Endpoints                   : 3.3.4.4 (VR-Default)
Network Ports [VXLAN]             : 1-64
Dynamic Virtual Networks          : On
* (CIT_30.4.0.524) FAserver2VPEX_1.215 # show fabric attach assignments
Fabric Attach Mode: Server
Port     VLAN  VLAN Name                         Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
5         100  SYS_VLAN_0100                     Dynamic   2800100  Active
5         101  SYS_VLAN_0101                     Dynamic   2800101  Active
```

```
# show fabric attach elements
Fabric Attach Mode: Server
                                                    Mgmt     Auto
System Id                    Port    Type           VLAN Tag Provision
---------------------------- ------- -------------- ---- --- --------------
00-04-96-9a-8d-45-00-01-00-18 5       Proxy (No Auth)  None Mix Disabled
00-11-88-fe-f3-66-00-01-00-1a 18      Proxy (No Auth)  None Mix Disabled
```

```
# show fdb
MAC                                   VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
--------------------------------------------------------------------------------------------
------------
00:04:96:98:a9:1a                   SYS_VLAN_0100(0100) 0005  d m        S   5
00:04:96:99:9e:43                   SYS_VLAN_0100(0100) 0000  s          SXE VR-
Default:2.3.44.5
00:04:96:9a:8d:6c                    SYS_BGP_0002(4082) 0000  s m        S   57
00:04:96:9a:8d:6c                             isc(3999) 0000  s m        S   57
00:04:96:9a:8d:b8                   SYS_VLAN_0100(0100) 0000  s          SXE VR-
Default:2.3.44.5
00:04:96:9a:8d:b8                    SYS_BGP_0003(4081) 0058  d m            49

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 6 Static: 4  Perm: 0  Dyn: 2  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
```

```
# show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    50.1.1.10        00:04:96:9a:8d:b8    0     YES  SYS_VLAN_0100  100   VR-
Default:2.3.44.5
VR-Default    50.1.1.100       00:04:96:99:9e:43    0     YES  SYS_VLAN_0100  100   VR-
Default:2.3.44.5
VR-Default    2.3.44.5         00:04:96:9a:8d:b8   20      NO  SYS_BGP_0003  4081  49
VR-Default    99.1.1.100       00:04:96:9a:8d:6c   20      NO  isc           3999  57
```

FA Server 2

```
# show fabric attach elements
Fabric Attach Mode: Server
                                                          Mgmt     Auto
System Id                      Port    Type               VLAN Tag Provision
------------------------------ ------- ------------------ ---- --- --------------
00-04-96-9a-8d-45-00-01-00-18 15      Proxy (No Auth)  None Mix Disabled
00-04-96-9a-8d-45-00-01-00-10 33      Proxy (No Auth)  None Mix Disabled
* (pacman debug) FAserver2VPEX_2.11 # show fabric attach assignments
Fabric Attach Mode: Server
Port     VLAN  VLAN Name                          Type     ISID/NSI  Status
-------  ----  --------------------------------  -------  --------  --------
15        100  SYS_VLAN_0100                     Dynamic   2800100  Active
15        101  SYS_VLAN_0101                     Dynamic   2800101  Active
```

```
# show virtual-network
Virtual Network                      Flags     Tenant VLAN
              Encap       ID     Encap Flags
================================================================================
SYS_VNET_02800100                    --S       SYS_VLAN_0100
              VXLAN    2800100   LRX
SYS_VNET_02800101                    --S       SYS_VLAN_0101
              VXLAN    2800101   LRX
================================================================================
Flags: (S) System Configured, (T) OTM Configured, (V) OVSDB Configured
Encap Flags: (L) Local Endpoints Configured,
       (R) Remote Endpoints Associated,
       (X) Exclude Tag
----------------------------------------
Total number of Virtual Networks  : 2
Local Endpoints                   : 3.3.4.4 (VR-Default)
Network Ports [VXLAN]             : 1-64
Dynamic Virtual Networks          : On
```

```
# show fdb
MAC                                     VLAN Name( Tag)  Age  Flags          Port /
Virtual Port List
-------------------------------------------------------------------------------------------
------------
00:04:96:98:a9:1a                   SYS_VLAN_0100(0100) 0076  d m        S   15
00:04:96:99:9e:43                   SYS_VLAN_0100(0100) 0000  s          SXE VR-
Default:2.3.44.5
00:04:96:9a:8d:b8                   SYS_VLAN_0100(0100) 0000  s          SXE VR-
Default:2.3.44.5
00:04:96:9a:8d:b8                    SYS_BGP_0003(4086) 0040  d m            61
00:04:96:9c:d4:5a                    SYS_BGP_0002(4087) 0000  s m        S   53
00:04:96:9c:d4:5a                             isc(3999) 0000  s m        S   53

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow, E - EVPN

Total: 6 Static: 4  Perm: 0  Dyn: 2  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
```

```
# show iparp
VR            Destination      Mac                Age  Static  VLAN          VID   Port
VR-Default    50.1.1.1         00:04:96:98:a9:1a    1      NO  SYS_VLAN_0100  100   15
```

```
VR-Default    50.1.1.10        00:04:96:9a:8d:b8    0     YES  SYS_VLAN_0100  100   VR-
Default:2.3.44.5
VR-Default    50.1.1.100       00:04:96:99:9e:43    0     YES  SYS_VLAN_0100  100   VR-
Default:2.3.44.5
VR-Default    2.3.44.5         00:04:96:9a:8d:b8   20      NO  SYS_BGP_0003  4086  61
VR-Default    99.1.1.1         00:04:96:9c:d4:5a   20      NO  isc           3999  53

Dynamic Entries  :           3             Static Entries            :          2
Pending Entries  :           0

ARP address check:    Enabled              ARP refresh               :    Enabled
Timeout          :          20 minutes     ARP Sender-Mac Learning   :   Disabled
Locktime         :        1000 milliseconds
Retransmit Time  :        1000 milliseconds
Reachable Time   :      900000 milliseconds (Auto)
Fast Convergence :          On

ARP Global Settings
Max Entries         :    21096
Max Pending Entries :      256
Max Proxy Entries   :      256
```

Fabric Attach Automatic LAG Configuration Example

ExtremeXOS Fabric Attach proxy received same Fabric TLVs from uplink VSP Fabric Attache servers.

VSP1---VSP2

\ /

ExtremeXOS Fabric Attach Proxy

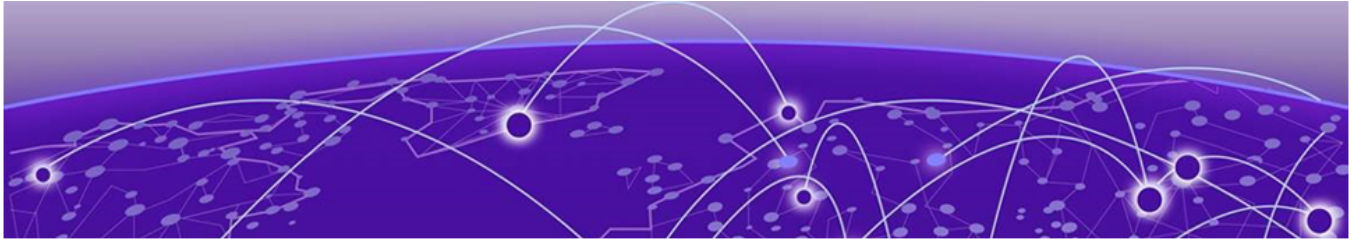
```
MLT/SMLT-id is 155
SystemId       : 00:01:00:00:01:cc
Connection Type: MLT
```

And so ExtremeXOS Fabric Attach proxy creates Fabric Attach Automatic LAG bundling on 1:57 and 1:58 ports.

```
# show ports 1:57 sharing
Load Sharing Monitor
Config    Current Agg     Min    Ld Share        Ld Share  Agg Link  Link Up
Master    Master  Control Active Algorithm Flags Group     Mbr State Transitions
================================================================================
  1:57   1:57     Static     1    L3        Ad    1:57      Y     A       5
                                 L3        d     1:58      Y     A       2
================================================================================
Link State: A-Active, D-Disabled, R-Ready, NP-Port not present, L-Loopback
Minimum Active: (<) Group is down. # active links less than configured minimum
Load Sharing Algorithm: (L2) Layer 2 address based, (L3) Layer 3 address based
                        (L3_L4) Layer 3 address and Layer 4 port based
                        (custom) User-selected address-based configuration
Custom Algorithm Configuration: ipv4 L3-and-L4, xor
Custom Hash Seed: Switch MAC address (0x96ED4C00)
Flags:
    A - All: Distribute to all members,
    d - Dynamically created shared port,
    L - Local Slot: Distribute to members local to ingress slot,
    P - Port Lists: Distribute to per-slot configurable subset of members,
```

```
    R - Resilient Hashing enabled.
Number of load sharing trunks: 6 (1 displayed)

# show lldp ports 1:57 1:58 neighbors detailed

--------------------------------------------------------------------------------
LLDP Port 1:57 detected 1 neighbor
  Neighbor: 00:04:96:B0:70:00/1/3, age 27 seconds
    - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:B0:70:00
    - Port ID type: ifName (5)
      Port ID     : "1/3"
    - Time To Live: 120 seconds
    - System Name: "VSP4900-CORE2"
    - System Description: "VSP-4900-24XE (8.2.0.0_B028) (PRIVATE)"
    - System Capabilities : "Bridge, Router"
      Enabled Capabilities: "Bridge, Router"
    - Management Address Subtype: IPv4 (1)
      Management Address        : 10.127.16.69
      Interface Number Subtype  : Unknown (1)
      Interface Number          : 0
      Object ID String          : "null"
    - Port Description: "Extreme Networks Virtual Services Platform VSP-4900\
                         -24XE - 10GbCX Port 1/3"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 4
      State          : 8
      Management Vlan: 0
      SystemId       : 00:01:00:00:01:cc
      Link Info      : 30-9b-00-9b
      Connection Type: MLT
      SMLT ID        : 155
      MLT ID         : 155
      EXOS Slot:Port : 1:57 (Master Port)

    - Extreme Fabric Attach NSI(I-SID)/VLAN map
    Status  VlanID  NSI(I-SID)
    ------  ------  --------
         2     101     10106
         2     200     20020


--------------------------------------------------------------------------------
LLDP Port 1:58 detected 1 neighbor
  Neighbor: 00:04:96:B0:68:00/1/9, age 28 seconds
    - Chassis ID type: MAC address (4)
      Chassis ID     : 00:04:96:B0:68:00
    - Port ID type: ifName (5)
      Port ID     : "1/9"
    - Time To Live: 120 seconds
    - System Name: "VSP4900-CORE1"
    - System Description: "VSP-4900-24XE (8.2.0.0_B028) (PRIVATE)"
    - System Capabilities : "Bridge, Router"
      Enabled Capabilities: "Bridge, Router"
    - Management Address Subtype: IPv4 (1)
      Management Address        : 10.127.16.68
      Interface Number Subtype  : Unknown (1)
      Interface Number          : 0
      Object ID String          : "null"
    - Port Description: "Extreme Networks Virtual Services Platform VSP-4900\
                         -24XE - 10GbCX Port 1/9"
    - Avaya/Extreme Fabric Attach element
      Element Type   : 4
      State          : 8
      Management Vlan: 0
```

```
    SystemId       : 00:01:00:00:01:cc
    Link Info      : 30-9b-00-9b
    Connection Type: MLT
    SMLT ID        : 155
    MLT ID         : 155
    EXOS Slot:Port : 1:57 (Master Port)

  - Extreme Fabric Attach NSI(I-SID)/VLAN map
  Status  VlanID  NSI(I-SID)
  ------  ------  --------
       2     101     10106
       2     200     20020
```



FDB

FDB Contents on page 826
How FDB Entries Get Added on page 826
How FDB Entries Age Out on page 827
FDB Entry Types on page 827
Managing the FDB on page 829
MAC-Based Security on page 832
Managing MAC Address Tracking on page 836
Displaying FDB Entries and Statistics on page 838

The FDB (forwarding database) chapter is intended to help you learn about forwarding databases, adding and displaying entries and entry types, managing the FDB, and MAC-based security. This chapter also provides information about MAC Address tracking. The switch maintains a forwarding database (FDB) of all MAC addresses received on all of its ports. It uses the information in this database to decide whether a frame should be forwarded or filtered.

Note
See command references for details of the commands related to the FDB.

FDB Contents

Each *FDB (forwarding database)* entry consists of:

- The MAC address of the device
- An identifier for the port and *VLAN (Virtual LAN)* on which it was received
- The age of the entry
- Flags

Frames destined for MAC addresses that are not in the FDB are flooded to all members of the VLAN.

How FDB Entries Get Added

The MAC entries that are added to the *FDB* are learned in the following ways:

- Source MAC entries are learned from ingress packets on all platforms. This is Layer 2 learning.
- On all switches, MAC entries can be learned at the hardware level.

- Virtual MAC addresses embedded in the payload of IP ARP packets can be learned when this feature is enabled.
- Static entries can be entered using the command line interface (CLI).
- Dynamic entries can be modified using the CLI.
- Static entries for switch interfaces are added by the system upon switch boot-up.

The ability to learn MAC addresses can be enabled or disabled on a port-by-port basis. You can also limit the number of addresses that can be learned, or you can lock down the current entries and prevent additional MAC address learning.

On SummitStacks with different ExtremeSwitching switch models, the FDB tables on some switches can be filled before the tables on other switches. In this situation, when a lower-capacity FDB table cannot accept FDB entries, a message appears that is similar to the following:

```
HAL.FDB.Warning> MSM-A: FDB for vlanID1 mac 00:00:03:05:15:04 was not added to slot 3 -
table full.
```

How FDB Entries Age Out

Software Aging Platforms

When a MAC is learned on a *VLAN*, an *FDB* entry is created for this MAC VLAN combination.Once an FDB entry is created, the aging counter in the `show fdb` output increases from 0 to “polling inteval”.

The hardware is checked every “polling interval” seconds to see if there is traffic flow from the given FDB entry.If there is a traffic flow from this MAC, the entry is refreshed and aging counter is reset to 0.

If there is no traffic from that FDB entry during this polling interval, the age gets incremented till the configured age time (configured by `configure fdb agingtime` *`seconds`*). The entry gets removed when there is no traffic flow from this FDB entry when the age count reaches the configured age time.

Polling interval = FDB aging time/4 (subject to the minimum and maximum values being 10 and 60 seconds respectively).

FDB Entry Types

Dynamic Entries

A dynamic entry is learned by the switch by examining packets to determine the source MAC address, *VLAN*, and port information.

The switch then creates or updates an *FDB* entry for that MAC address. Initially, all entries in the database are dynamic, except for certain entries created by the switch at boot-up.

Entries in the database are removed (aged-out) if, after a period of time (aging time), the device has not transmitted. This prevents the database from becoming full with obsolete entries by ensuring that when a device is removed from the network, its entry is deleted from the database.

The aging time is configurable, and the aging process operates on the supported platforms as follows:

- On all platforms, you can configure the aging time to 0, which prevents the automatic removal of all dynamic entries.
- On all platforms, the aging process takes place in software and the aging time is configurable.

For more information about setting the aging time, see Configuring the FDB Aging Time on page 831.

Note
If the FDB entry aging time is set to 0, all dynamically learned entries in the database are considered static, non-aging entries. This means that the entries do not age, but they are still deleted if the switch is reset.

Dynamic entries are flushed and relearned (updated) when any of the following take place:

- A VLAN is deleted.
- A VLAN identifier (VLANid) is changed.
- A port mode is changed (tagged/untagged).
- A port is deleted from a VLAN.
- A port is disabled.
- A port enters blocking state.
- A port goes down (link down).

A non-permanent dynamic entry is initially created when the switch identifies a new source MAC address that does not yet have an entry in the FDB. The entry can then be updated as the switch continues to encounter the address in the packets it examines. These entries are identified by the “d” flag in the `show fdb` command output.

Static Entries

A static entry does not age and does not get updated through the learning process.

A static entry is considered permanent because it is retained in the database if the switch is reset or a power off/on cycle occurs. A static entry is maintained exactly as it was created. Conditions that cause dynamic entries to be updated, such as *VLAN* or port configuration changes, do not affect static entries.

To create a permanent static *FDB* entry, see Adding a Permanent Unicast Static Entry on page 829.

If a duplicate MAC address is learned on a port other than the port where the static entry is defined, all traffic from that MAC address is dropped. By default, the switch does not report duplicate addresses. However, you can configure the switch to report these duplicate addresses as described in Managing Reports of Duplicate MAC Addresses for Static Entries on page 831.

A locked static entry is an entry that was originally learned dynamically, but has been made static (locked) using the MAC address lock-down feature. It is identified by the "s," "p," and "l" flags in show fdb command output and can be deleted using the `delete fdb` command. See MAC Address Lockdown on page 1551 for more information about this feature.

Note
Static FDB entries created on *EAPS (Extreme Automatic Protection Switching)*- or *STP (Spanning Tree Protocol)*-enabled ports forward traffic irrespective of the port state. Consequently, you should avoid such a configuration.

Blackhole Entries

A blackhole entry configures the switch to discard packets with a specified MAC destination address.

Blackhole entries are useful as a security measure or in special circumstances where a specific source or destination address must be discarded. Blackhole entries can be created through the CLI, or they can be created by the switch when a port's learning limit has been exceeded.

Blackhole entries are treated like permanent entries in the event of a switch reset or power off/on cycle. Blackhole entries are never aged out of the database.

Private VLAN Entries

A Private *VLAN* (PVLAN) creates special *FDB* entries. These are described in MAC Address Management in a PVLAN on page 2236.

Managing the FDB

Increasing the FDB Table Size

Some platforms provide a Unified Forwarding Table that allows for flexible allocation of entries to L2, L3, or routes. You can configure this table with the `configure forwarding internal-tables [ 12-and-l3 | more [l2 | l3-and-ipmc | routes {ipv6-mask-length [64 | 128]}]]` command.

Adding a Permanent Unicast Static Entry

To add a static entry use the following command:

```
create fdb mac_addr vlan vlan_name [ports port_list | blackhole]
```

The following example adds a permanent static entry to the *FDB*:

```
create fdb 00:E0:2B:12:34:56 vlan marketing port 3:4
```

The permanent entry has the following characteristics:

- MAC address is 00:E0:2B:12:34:56.
- *VLAN* name is marketing.
- Slot number for this device is 3 (SummitStacks only).
- Port number for this device is 4.

You can specify multiple ports when you create a unicast static entry. However, all ports in the list must be on the same SummitStack switch. When the port list contains ports on different slots, the following error is generated:

```
Error: Multiple ports must be on the same slot for unicast MAC FDB entries.
```

Once the multiport static FDB entry is created, any ingress traffic with a destination MAC address matching the FDB entry is multicasted to each port in the specified list. If the FDB entry is the next hop for an IP adjacency, unicast routing sends the packet to the first port in the list.

Note
When a multiport list is assigned to a unicast MAC address, load sharing is not supported on the ports in the multiport list.

ExtremeSwitching series switches do not support this multiport feature natively using the FDB table. Instead, for each FDB entry of this type, a series of system *ACL (Access Control List)*s have been installed which match the specified MAC address and VLAN ID, and override the egress port forwarding list with the supplied list of ports. Multiple ACLs per FDB are required to handle Layer 2 echo kill by installing a unique ACL per individual port in the list to send matching traffic to all other ports in the list.

User-configured ACLs take precedence over these FDB-generated ACL rules, and the total number of rules is determined by the platform.

The hardware ACL limitations for each platform are described in ACLs on page 341.

Adding a Permanent Multicast Static Entry

On SummitStack and ExtremeSwitching series switches, you can create *FDB* entries to multicast MAC addresses (that is, 01:00:00:00:00:01) and list one or more ports.

Use the `create fdb mac_addr vlan vlan_name [ports port_list | blackhole]` command to enter the multicast FDB address. After traffic with a multicast MAC destination address enters the switch, that traffic is multicast to all ports on the list.

However, if the MAC address is in the IP multicast range (for example, 01:00:5e:XX:XX:XX), *IGMP (Internet Group Management Protocol)* snooping rules take precedence over the multicast static FDB entry. Of course, if you disable IGMP snooping on all VLANs, the static FDB entry forwards traffic.

Configuring the FDB Aging Time

- To configure the aging time for dynamic *FDB* entries, use the following command:.

  ```
  configure fdb agingtime seconds
  ```

  If the aging time is set to 0, all aging entries in the database are defined as static, nonaging entries. This means the entries will not age out, but non-permanent static entries can be deleted if the switch is reset.
- To display the aging time, use the following command:.

  ```
  show fdb
  ```

Adding Virtual MAC Entries from IP ARP Packets

Generally, the *FDB* is programmed with the source MAC address of frames that contain an IP ARP payload. MAC entries present in the ARP payload as Sender-MAC are not learned. When IP ARP Sender-MAC learning is enabled, the switch learns both the source MAC address and the Sender-MAC from the ARP payload, and the switch programs these MAC addresses in the FDB.

This feature is useful when you want the switch to learn the Sender-MAC address for a redundant protocol, such as *VRRP (Virtual Router Redundancy Protocol)*. For example, if your network has a gateway with a virtual MAC address, the switch learns the system MAC address for the gateway. If you enable the IP ARP Sender-MAC learning feature, the switch also learns the virtual MAC address embedded in IP ARP packets for the gateway IP address.

- To enable the IP ARP sender-MAC learning feature, use the command:

  ```
  enable learning iparp sender-mac
  ```
- To view the configuration of this feature, use the command:

  ```
  show iparp
  ```
- To disable this feature, use the command:

  ```
  disable learning iparp sender-mac
  ```

Managing Reports of Duplicate MAC Addresses for Static Entries

By default, if a MAC address that is a duplicate of a static MAC address entry is learned on another port (other than the port where the static MAC address is configured), traffic from the duplicate address is silently dropped.

- To enable or disable EMS and *SNMP (Simple Network Management Protocol)* reporting of duplicate addresses for static entries, use the commands:

  ```
  enable fdb static-mac-move
  ```

  ```
  disable fdb static-mac-move
  ```
- To control the number of EMS and SNMP reports per second issued, use the commands:

  ```
  configure fdb static-mac-move packets count
  ```
- To display the configuration of this feature, use the commands:

  ```
  show fdb static-mac-move configuration
  ```

Clearing FDB Entries

You can clear dynamic and permanent entries using different CLI commands. Clear dynamic *FDB* entries by targeting:

- Specified MAC addresses
- Specified ports
- Specified VLANs
- All blackhole entries
- To clear dynamic entries from the FDB, use the command:
  `clear fdb {mac_addr | ports port_list | vlan vlan_name | blackhole}`
- To clear permanent entries from the FDB, use the command:
  `delete fdb [all | mac_address [vlan vlan_name ]`

Supporting Remote Mirroring

The remote mirroring feature copies select traffic from select ports and VLANs and sends the copied traffic to a remote switch for analysis.

The mirrored traffic is sent using a *VLAN* that is configured for this purpose. For more information, see MLAG Limitations and Requirements on page 295.

Transit switches are the switches between the source switch where ports are mirrored and the destination switch where the mirrored traffic exits the network to a network analyzer or network storage device.

Because the mirrored traffic is an exact copy of the real traffic, a transit switch can learn the MAC addresses and make incorrect forwarding decisions.

MAC-Based Security

MAC-based security allows you to control the way the *FDB* is learned and populated. By managing entries in the FDB, you can block and control packet flows on a per-address basis.

MAC-based security allows you to limit the number of dynamically-learned MAC addresses allowed per virtual port. You can also “lock” the FDB entries for a virtual port, so that the current entries will not change, and no additional addresses can be learned on the port.

You can also prioritize or stop packet flows based on the source MAC address of the ingress *VLAN* or the destination MAC address of the egress VLAN.

Note
For detailed information about MAC-based security, see Security on page 1628.

Managing MAC Address Learning

By default, MAC address learning is enabled on all ports. MAC addresses are added to the *FDB* as described in How FDB Entries Get Added on page 826.

When MAC address learning is disabled on a port, the switch no longer stores the source address information in the FDB. However, the switch can still examine the source MAC address for incoming packets and either forward or drop the packets based on this address. The source address examination serves as a preprocessor for packets. Forwarded packets are forwarded to other processes, not to other ports. For example, if the switch forwards a packet based on the source address, the packet can still be dropped based on the destination address or the egress flooding configuration.

When MAC address learning is disabled, the two supported behaviors are labeled as follows in the software:

- forward-packets
- drop-packets

The drop-packets behavior is supported on SummitStack and ExtremeSwitching series switches. When the drop-packets option is chosen, EDP packets are forwarded, and all unicast, multicast, and broadcast packets from a source address not in the FDB are dropped. No further processing occurs for dropped packets.

The disable learning forward-packets option saves switch resources (FDB space), however, it can consume network resources when egress flooding is enabled. When egress flooding is disabled or the drop-packet option is specified, disabling learning adds security by limiting access to only those devices listed in the FDB.

Note

When the forward-packet option is chosen,

- If unicast, multicast, and broadcast packet from a source address is not present in the FDB , the packets is flooded.
- If the destination MAC is present in the forwarding database, the packet is forwarded.

- To disable learning on specified ports, use the command:

  `disable learning {`**`drop-packets`** `|` **`forward-packets`**`} port [`*`port_list`* `|` **`all`**`]`

- To enable learning on specified ports, use the command:

  `enable learning {`**`drop-packets`**`} ports [`**`all`** `|` *`port_list`*`]`

Managing Egress Flooding

Egress flooding takes action on a packet based on the packet destination MAC address. By default, egress flooding is enabled, and any packet for which the destination address is not in the *FDB* is flooded to all ports except the ingress port.

You can enhance security and privacy as well as improve network performance by disabling Layer 2 egress flooding on a port, *VLAN*, or VMAN. This is particularly useful

when you are working on an edge device in the network. Limiting flooded egress packets to selected interfaces is also known as upstream forwarding.

Note
Disabling egress flooding can affect many protocols, such as IP and ARP.

Figure 90 illustrates a case where you want to disable Layer 2 egress flooding on specified ports to enhance security and network performance.

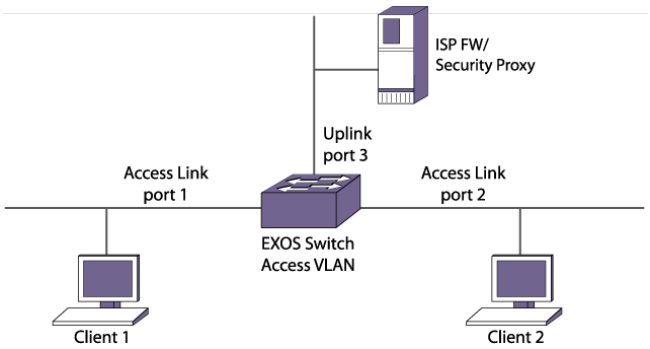


Figure 90: Upstream Forwarding or Disabling Egress Flooding Example

In this example, the three ports are in an ISP-access VLAN. Ports 1 and 2 are connected to clients 1 and 2, respectively, and port 3 is an uplink to the ISP network. Because clients 1 and 2 are in the same VLAN, client 1 could possibly learn about the other client's traffic by sniffing client 2's broadcast traffic; client 1 could then possibly launch an attack on client 2.

However, when you disable all egress flooding on ports 1 and 2, this sort of attack is impossible, for the following reasons:

- Broadcast and multicast traffic from the clients is forwarded only to the uplink port.
- Any packet with unlearned destination MAC addresses is forwarded only to the uplink port.
- One client cannot learn any information from the other client. Because egress flooding is disabled on the access ports, the only packets forwarded to each access port are those packets that are specifically targeted for one of the ports. There is no traffic leakage.

In this way, the communication between client 1 and client 2 is controlled. If client 1 needs to communicate with client 2 and has that IP address, client 1 sends out an ARP request to resolve the IP address for client 2.

Guidelines for Enabling or Disabling Egress Flooding

The following guidelines apply to enabling and disabling egress flooding:

- Egress flooding can be disabled on ports that are in a load-sharing group. In a load-sharing group, the ports in the group take on the egress flooding state of the master port; each member port of the load-sharing group has the same state as the master port.
- *FDB* learning takes place on ingress ports and is independent of egress flooding; either can be enabled or disabled independently.

- Disabling unicast (or all) egress flooding to a port also prevents the flooding of packets with unknown MAC addresses to that port.
- Disabling broadcast (or all) egress flooding to a port also prevents the flooding of broadcast packets to that port.
- For SummitStack and ExtremeSwitching series switches, the following guidelines apply:
  - You can enable or disable egress flooding for unicast, multicast, or broadcast MAC addresses, as well as for all packets on one or more ports.
  - Disabling multicasting egress flooding does not affect those packets within an *IGMP* membership group at all; those packets are still forwarded out.
  - If IGMP snooping is disabled, multicast packets with static FDB entries are forwarded according to the FDB entry.

Configuring Egress Flooding

To enable or disable egress flooding on SummitStack and the ExtremeSwitching series switches, use the following commands:

```
enable flooding [all_cast | broadcast | multicast | unicast] ports
[port_list | all]
```

```
disable flooding [all_cast | broadcast | multicast | unicast] ports
[port_list | all]
```

Displaying Learning and Flooding Settings

To display the status of MAC learning and egress flooding, use the following commands:

```
show ports {mgmt | port_list | tag tag} information {detail}
```

```
show vlan {virtual-router vr-name}
```

```
show vman vman_list
```

The flags in the command display indicate the status.

Creating Blackhole FDB Entries

A blackhole *FDB* entry discards all packets addressed to or received from the specified MAC address. A significant difference between the above *ACL* policy and the `create fdb` command blackhole option is the hardware used to implement the feature. Platforms with limited hardware ACL table sizes are able to implement this feature using the FDB table instead of an ACL table.

- To create a blackhole FDB entry, use the command:

  ```
  create fdb mac_addr vlan vlan_name [ports port_list | blackhole]
  ```

  There is no software indication or notification when packets are discarded because they match blackhole entries.

The blackhole option is also supported through access lists.

Note
Blackhole is not supported on port-specific *VLAN* tags.

For example, the following ACL policy would also blackhole traffic destined to or sourced from a specific MAC address:

```
entry blackhole_dest {
if {
ethernet-destination-address 00:00:00:00:00:01;
} then {
deny;
}
}
entry blackhole_source {
if {
ethernet-source-address 00:00:00:00:00:01;
} then {
deny;
}
}
```

Managing MAC Address Tracking

The MAC address tracking feature tracks *FDB* add, move, and delete events for specified MAC addresses and for specified ports.

When MAC address tracking is enabled for a port, this feature applies to all MAC addresses on the port.

When an event occurs for a specified address or port, the software generates an EMS message and can optionally send an *SNMP* trap. When MAC address tracking is enabled for a specific MAC address, this feature updates internal event counters for the address. You can use this feature with the Universal Port feature to configure the switch in response to MAC address change events (for an example, see Universal Port on page 2135).

Note
When a MAC address is configured in the tracking table, but detected on a MAC tracking enabled port, the per MAC address statistical counters are not updated.

The MAC address tracking feature is always enabled; however, you must configure MAC addresses or ports before tracking begins. The default configuration contains no MAC addresses in the MAC address tracking table and disables this feature on all ports.

Adding and Deleting MAC Addresses for Tracking

Use the following commands to add or delete MAC addresses in the MAC address tracking table:

```
create fdb mac-tracking entry mac_addr
```

```
delete fdb mac-tracking entry [mac_addr | all]
```

Enabling and Disabling MAC Address Tracking on Ports

Use the following command to enable or disable MAC addresses tracking on specific ports:

```
configure fdb mac-tracking {[add|delete]} ports [port_list|all]
```

Enabling and Disabling SNMP Traps for MAC Address Changes

The default switch configuration disables *SNMP* traps for MAC address changes.

Use the following commands to enable or disable SNMP traps for MAC address tracking events:

```
enable snmp traps fdb mac-tracking
```

```
disable snmp traps fdb mac-tracking
```

Configuring Automatic Responses to MAC Tracking Events

The EMS messages produced by the MAC address tracking feature can be used to trigger Universal Port profiles. These are described in Event Management System Triggers on page 2141.

The subcomponent name for MAC address tracking events is FDB.MACTracking.

Displaying the Tracked MAC Addresses and Tracking Statistics

- To display the MAC address tracking feature configuration, including the list of tracked MAC addresses, use the command:

  ```
  show fdb mac-tracking configuration
  ```

- To display the counters for MAC address add, move, and delete events, use the command:

  ```
  show fdb mac-tracking statistics {mac_addr} {no-refresh}
  ```

Clearing the Tracking Statistics Counters

There are several ways to clear the MAC tracking counters:

- Use the `clear counters` command.
- Use the **0** key while displaying the counters with the `show fdb mac-tracking statistics {mac_addr}` command.
- Enter the `clear counters fdb mac-tracking [mac_addr | all]` command.

Displaying FDB Entries and Statistics

Display FDB Entries

- Display *FDB* entries using the following command:

  ```
  show fdb {blackhole {netlogin [all | mac-based-vlans]} | netlogin
  [all | mac-based-vlans] | permanent {netlogin [all | mac-based-vlans]}
  | mac_addr {netlogin [all | mac-based-vlans]} | ports port_list
  {netlogin [all | mac-based-vlans]} | vlan vlan_name | vlan_list
  {netlogin [all | mac-based-vlans]} | {{vpls} {vpls_name}}}
  ```

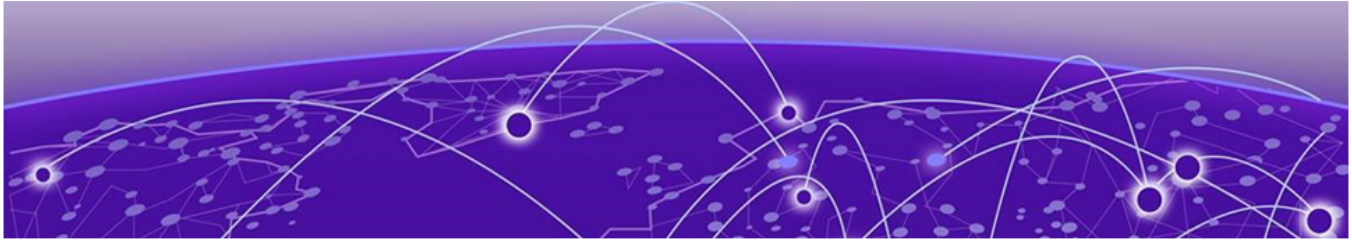
  With no options, this command displays all FDB entries.

Display FDB Statistics

To display *FDB* statistics, use the command:

```
show fdb stats {{ports {all | port_list} | vlan {all} | {vlan} vlan_name
| vlan_list } {no-refresh}}
```

With no options, this command displays summary FDB statistics.



Identity Management

Identity Management Overview on page 839
Identity Management Feature Limitations on page 857
Configuring Identity Management on page 858
Managing the Identity Management Feature on page 865
Displaying Identity Management Information on page 866

This chapter offers detailed information about the ExtremeXOS Identity Management feature. It provides an overview, as well as specific information on how to configure, manage and monitor this feature.

Identity Management Overview

The identity management feature allows you to learn more about the users and devices (such as phones and routers) that connect to a switch. In this chapter, users and devices are collectively called *identities*. The Identity Management feature:

- Captures identity information when users and devices connect to and disconnect from the switch.
- Stores captured identity information and identity event data in a local database.
- Generates EMS messages for user and device events.
- Makes collected identity information available for viewing by admin-level users and to management applications through XML APIs.
- Uses locally collected identity information to query an LDAP server and collect additional information about connected identities.
- Supports custom configurations called *roles*, which are selected based on identity information collected locally and from an LDAP server.
- Uses roles to enable traffic filtering, counting, and metering on ports (using ACLs and policies) in response to identity events (connections, disconnections, and time-outs).
- Supports the configuration of blacklist to deny all access to an identity and whitelists to permit all access to an identity.
- Supports the configuration of greylist to enable the network administrator to choose usernames whose identity is not required to be maintained. When these usernames are added to greylist, the Identity Management module does not create an identity when these users log on.

- Integrates with UPM to modify the switch configuration in response to discovered identities.
- Services users under different domains by allowing different domains to be configured and then associating different LDAP servers for those different domains.

Note
IDM and ONEPolicy are not supported together and it is not recommended to enable both, since handling rule/role-based actions is not supported, except to support Kerberos Authentication with NAC as a RADIUS server and can be used in conjunction with IDM XML event triggers.

Note
When using IDM commands, you should generally avoid the encrypted option. Passwords provided in commands in plain text are saved in encrypted format.

Identity Information Capture

The identity management feature collects user and device data whenever users or devices connect to or disconnect from the switch. Table 45 lists the identity management attributes that the identity manager process collects from the listed switch software components.

Table 45: Identity (User/Device) Attributes and Source Software Components

Attribute	NetLogin	LLDP	FDB	IP-Security	Kerberos Snooping
User's MAC address	X	X	X	X	X
Authentication and unauthentication time stamp	X	X	X	X	X
User's port	X	X	X	X	X
User's VLANs	X		X	X	X
User's identity	X	X			X
IPv4 to MAC binding			X	X	X
NetLogin authentication protocol	X				
Authentication failures	X				
Device capabilitiesa		X			
Device model name[a]		X			
Device manufacturer name[a]		X			

[a] Identity manager receives these attributes only from LLDP enabled ports when the remote device is configured to send the corresponding TLV.

The software components in Table 45 trigger identity attribute collection when a user or device connects to the switch. All components provide the MAC address, authentication and unauthentication time stamps, and the port to which the identity connected. When multiple components are triggered by a user or device connection, the triggers usually happen at different times. Identity manager responds to all identity event triggers, adding additional information to the identity database each time it becomes available.

To capture all the available attribute information listed in Table 45, enable the following features:

- Network Login
- LLDP
- IP Security

By default, the identity management feature collects information from all devices connected to identity management enabled ports which does Kerberos authentication using Kerberos snooping. Kerberos authentication, or ticketing, is used by Microsoft's Active Directory. The Kerberos snooping feature collects identity attributes from Kerberos Version 5 traffic. This feature does not capture information from earlier versions of Kerberos.

Note
We recommend that you enable CPU DoS protect in combination with Kerberos snooping to make sure the CPU is not flooded with mirrored Kerberos packets in the event of a DoS attack on Kerberos TCP/UDP ports. If the rate limiting capability is leveraged on capable platforms, it is applied on CPU mirrored packets.

Because an identity entry in the identity manager database can contain information from various software components (listed in Table 45 on page 840), when a component other than a network login triggers an identity removal, only the attributes supplied by that component are removed from the identity. When network login triggers an identity removal, all attributes for that identity are removed from the identity manager database.

Identity Names

After identity attributes are captured, they can be viewed with show commands on the switch. The identity ID Name assigned to each identity depends on the identity attributes collected. For example, if a MAC address detected by *FDB (forwarding database)* is not correlated by at least one other software component, the identity is considered an unknown identity, and identity manager creates an identity entry with the name unknown_<MAC-Address>, where MAC-Address is replaced with the actual MAC address.

When an FDB detected MAC address is correlated by another software component, the identity is considered a known identity, and the identity manager names the identity based on the identity attributes.

For example, if a user name is collected, the user name becomes the ID name. If a username is not discovered, identity manager creates a name based on the MAC address.

Identity manager can change the ID name when additional attributes are learned, or when the identity status changes between known and unknown. For example, if *LLDP (Link Layer Discovery Protocol)* sends an identity removal trigger to the identity manager for an LLDP-based identity, and if a valid FDB entry exists for the removed identity, the identity manager reestablishes the identity as an unknown identity (unknown_<MAC-Address>).

Note

If FDB triggers the removal of the MAC address for an unknown identity, the identity manager deletes the corresponding unknown identity after a period of time.

Kerberos Authentication Type Support

Kerberos authentication support in ONEPolicy is achieved using NAC with IDM XML events.

With *NetLogin*, ONEPolicy, and IDM enabled, once the MAC address is authenticated and IDM table is populated with the MAC user and with Kerberos correlated user, using XML target configured in ExtremeXOS, the IDMGR events are be sent to Extreme Management Center server using HTTP/HTTPS. Extreme Management Center after receiving the XML event decides what to do with the profiles configured for Kerberos.

With NetLogin, ONEPolicy, and IDM enabled, once the MAC address is authenticated using NAC as a RADIUS Server and the IDM table is populated with the MAC user and with Kerberos correlated user, using XML target configured in ExtremeXOS switch, the IDMGR events are sent to Extreme Management Center server (Netsight) using HTTP/HTTPS. For ONEPolicy + IDM + Kerberos to work NAC should be the RADIUS server. Using an external RADIUS server, and with XML events alone configured to send to Extreme Management Center, does not work since the IDM table is not populated with ONEPolicy enabled for NetLogin MAC/Dot1x/Web entries. IDM table is populated only for Kerberos user with ONEPolicy enabled.

For information about how to configure Kerberos authentication type, see Configuring Kerberos Authentication Type on page 865.

Application of ACLs and Policies for Identities

Each time the identity manager detects a new identity or an identity change, it evaluates the identity attributes to determine which role to apply to the identity. A role is a switch configuration entity that identifies ACLs to apply to a port in response to an identity presence.

How Roles Affect Ports

A role is a configuration entity to which you can add multiple policy files or dynamic *ACL (Access Control List)* rules. When an identity is matched to a role, any policies or rules attached to that role are applied to the port to which the identity connected. These rules or policies permit or deny traffic, increment traffic counters, or implement traffic meters. When identity manager detects a removal trigger for an identity, all rules or policies associated with the identity are removed from the port on which the identity was detected.

Authenticated and Unauthenticated Roles

The identity management feature supports two default roles—authenticated and unauthenticated. No default rules or policies are configured for these roles, but you can add rules or policies to these roles.

Authenticated identities are known identities that meet the following requirements:

- Are not included in the blacklist or whitelist.
- Do not meet the match criteria for any user-defined roles.
- Cannot meet the match criteria for any user-defined role with LDAP attributes because no LDAP server is available or because LDAP queries are disabled.
- Are detected either through network login (using any of the network login methods) or through Kerberos snooping.

The unauthenticated role applies to all identities that do not match any other default or user-defined role.

For example, the following identities are placed in the unauthenticated role:

- A device detected by *LLDP* that has not authenticated through network login and does not match any other default or user-defined role.
- A user who does not successfully log in using Kerberos login and does not match any other default or user-defined role.
- A device discovered through IP ARP or *DHCP (Dynamic Host Configuration Protocol)* snooping that does not match any other default or user-defined role.
- Any identity classified as an unknown identity.

Note

The unauthenticated role is not applied to network login and Kerberos users because those users are either authenticated or denied by network login.

One option for configuring the unauthenticated role policy/rule is to allow DNS, DHCP, and Kerberos traffic, and deny all other traffic. This configuration allows identities to attempt log in, and denies access to identities that do not successfully log in.

Blacklist and Whitelist Roles

Blacklist and whitelist roles are special roles that are evaluated before all the other role types. If an identity is listed in a blacklist, that identity is denied all access to the network without consideration of any other roles to which it might belong. Similarly, if

a discovered identity is found in the whitelist, that identity is granted complete network access, and no further role processing occurs for that identity.

You can configure identities in a blacklist or whitelist using any one the following identity attributes:

- MAC address
- IPv4 address
- Username (with or without a domain name)

The type of identity attribute specified in a blacklist or whitelist impacts the locations from which an identity can access a switch. For example, if a MAC address or an IP address is specified in a blacklist, no access is permitted from any user at devices with the specified address. If a username is specified in a whitelist, that user is permitted access from all locations.

When an identity accesses the switch and that identity is in a blacklist or whitelist, the switch installs a specific deny or allow *ACL* on the port through which the identity attempts access. The installed ACL is an active ACL that explicitly denies or allows traffic from that identity. There is no passive action that takes place if the identity is not listed in the ACL. When the identity is not listed in a blacklist or whitelist, the switch checks for matches to other roles as described in Role Precedence and Priority on page 850.

Greylist Roles

Greylist feature enables the network administrator to choose usernames whose identity is not required to be maintained. When these usernames are added to greylist, the Identity Management module does not create an identity when these users log on.

This will be useful in a scenario wherein multiple users log in from same device at the same time. For example, actual user has logged into computer after Kerberos authentication. Later, Anti-Virus Agent (AVAgent) software starts within the same computer and does Kerberos authentication.

This will result in losing actual user identity and creating identity for AVAgent. Configuring AVAgent's username in greylist will prevent the above situation and actual user identity along with policies will be retained when AVAgent user logs in.

List Precedence Configuration

Greylist entries have higher precedence over blacklist and whitelist entries by default. This means that IDM consults with greylist first, upon detection of user, and then decides if the identity needs to be created. If there is no matching greylist entry, IDM proceeds with role identification for the user. However, greylist precedence is configurable. The following are three possibilities for greylist precedence configuration:

- greylist, blacklist, whitelist
- blacklist, greylist, whitelist
- blacklist, whitelist, greylist

At this time, blacklist always has precendence over whitelist. To change list precedence, disable IDM first. Disabling IDM is required since reverting roles and revoking policies due to greylist entries may increase processing load. When precedence configuration

is changed, each entry present in the list with lower precedence (new precedence) is checked with each entry present in all the lists with higher precedence. If any existing entry becomes ineffective, details of those entries are displayed at the CLI prompt.

User-Defined Roles

User-defined roles allow you to create custom roles that can restrict, count, and meter traffic for identities you want to control. CLI commands allow you to do the following:

- Create a user defined role.
- Configure identity match criteria that determine which identities use a role.
- Add dynamic *ACL* rules or policies to a role so that those policies are applied to ports to which a matching identity connects.
- Assign a priority level to each role to determine which role applies when multiple roles are matched to an identity.
- Establish hierarchical roles that can be used to support topologies built around a company organization structure or a geographical layout.

When specifying match criteria for a role, you can specify identity attributes collected by identity manager (see Identity Information Capture on page 840) and those collected from an LDAP server. When configured for an LDAP server, identity manager can send a query to the server with locally collected attributes and retrieve additional attributes for the identity, such as an employee department or title. The use of an LDAP server allows you to design roles that serve departments or localities.

Identity Attributes on an LDAP Server

When identity manager is configured to connect to an LDAP server, identity manager can query the server for the identity attributes listed in Table 46.

Table 46: LDAP Attributes for Role Selection

Attribute	Active Directory LDAP Attribute	Attributes Allowed in Identity Manager Match Criteria
City	l	l or location
State	st	st or state
Country	co	co or country
Employee ID	employeeID	employeeID
Title	title	title
Department	department	department
Company	company	company
Email Address	mail	mail or email

An LDAP query contains one or more of the identity attributes listed in Table 45 on page 840.

If an LDAP server fails to respond, the next configured LDAP server is contacted. When a server query succeeds, all further LDAP queries are sent to that LDAP server. All LDAP

servers should be configured to synchronize the user information available in each of them.

Note
Identity manager supports a maximum of eight LDAP servers.

Match Criteria for Selecting User-Defined Roles

When you create a user-defined role, you must define the match criteria that determines which identities will use the role. The match criteria is a group of identity attributes and the attribute values that must match identities to which this role is assigned. For example, the following command creates a role named US-Engr that matches employees whose title is Engineer and who work in United States:

```
create identity-management role US-Engr match-criteria "title contains Engineer; AND
country == US;" priority 100
```

The match criteria are a series of attributes, operators, and attribute values, all of which are described in the Switch Engine v33.6.1 Command References . Each role can define up to 16 attribute requirements, and there are additional operators such as not equal. Beginning in ExtremeXOS 15.3, the match criteria attributes are combined using the AND or OR operators, not a combination of both. When multiple roles are matched to an identity, the role with the highest priority (lowest numerical value) applies.

In the preceding example, identity manager must be configured to query an LDAP server because the identity attributes listed in the match criteria are not discovered locally.

The match criteria for a role establish the role as on of two types:

- Local user-defined role
- LDAP user-defined role

A local user-defined role uses only the following locally discovered attributes (which are listed in the following table) in the match criteria:

- User's MAC address
- MAC OUI
- User's port
- User's identity
- IPv4-to-MAC binding
- Device capabilities
- Device model name
- Device manufacturer name

Because a local user-defined role does not require LDAP provided attributes, the role can be matched to identities when an LDAP server is unavailable, or when LDAP processing is disabled for network login authenticated identities. A local user-defined role can serve as a backup role to an LDAP user-defined role.

An LDAP user-defined role uses one or more of the LDAP attributes listed in Identity Attributes on an LDAP Server on page 845 in the match criteria, and it can also use the attributes listed in Identity Information Capture on page 840. An LDAP user-defined role gives you more flexibility in selecting attributes for the match criteria. However, if no LDAP server is available, and the identity attributes do not match a local user-defined role, one of the two default roles is applied to the identity.

Role Policy Order

Previously, the policy or dynamic rule associated to the role occurred in the order of configuration. There was no way for you to change the order of the policy or dynamic rule associated with the role. ExtremeXOS 15.2 supported the ability to change the order of the policy or dynamic rule associated with the role. You can also change the order of the policy or dynamic rule during the run time. Even if the role is assigned to some identities, the policy or the dynamic rule associated to the role can be changed.

Role Hierarchy

To make it easier to manage users, the role management feature supports hierarchical roles. Hierarchical roles can be defined to reflect different organizational and functional structures. To create a role hierarchy, one or more roles are defined as child roles of what becomes a parent role. Each role can have a maximum of eight child roles and only one parent role. This feature can support up to five levels of parent and child roles. With respect to role hierarchy and match criteria, there is no restriction across roles. Beginning in 15.3 release, a user can have the parent role with AND, and the child role with OR, or vice versa. The inheritance of match criteria to the child role from the parent role uses AND as in previous releases.

Role Inheritance

Child roles inherit the policies of the parent role and all roles above the parent in the hierarchy. When an identity is assigned to a role, the policies and rules defined by that role and all higher roles in the hierarchy are applied.

When the parent role is deleted or when the parent-child relationship is deleted, the child role no longer inherits the parent role's policies and the policies are immediately removed from all identities mapped to the child role.

Because the maximum role hierarchy depth allowed is five levels, the maximum number of policies and dynamic ACLs that can be applied to a role is 40 (five role levels x eight policies/rules per role). Figure 91 on page 849 shows an example of hierarchical role management.

Match Criteria Inheritance

Beginning in release 15.2, the child role can inherit the match criteria of the parent role.

This means that the match criteria does not need to be duplicated in all levels of hierarchy.

For example, if you have roles called Employee, India employee, and India engineer in a hierarchy, previously the match criteria of the three roles would have been:

```
“company == Extreme”
“company == Extreme; AND country == India”
“company == Extreme; AND country == India; AND department = Engineering”
```

This can be simplified into the following since the child role automatically inherits the parent role’s match criteria:

```
“company == Extreme”
“country == India”
“department = Engineering”
```

Once this support is enabled, user identity must satisfy not only the role’s match criteria, but its parent and ancestors also. This support can be enabled/disabled using CLI or XML. You no longer have to repeat the match criteria configured in the parent role in the child roles also.

Note

- Role match criteria inheritance can only be enabled if all of the existing roles have higher priority than their descendants. If this condition is not satisfied, match criteria inheritance will fail.
- Once this feature is enabled, you cannot configure a child role with lesser priority (higher priority number) than its parent.
- Enabling this feature changes the order of the roles according to the parent-child relationship.
- Incoming identities are matched against the child role and then against the parent irrespective of the order of creation.

For example, Role A and Role B have match criteria MC-A and MC-B, respectively. Role B is a child role of Role A. When match criteria inheritance is disabled, an identity matches Role B criteria, and then it is placed under Role B with no further check.

When match criteria inheritance is enabled, the same identity, after satisfying Role B’s match criteria, is then checked against Role A’s match criteria. Once the identity satisfies child and parent match criteria, it is placed under Role B.

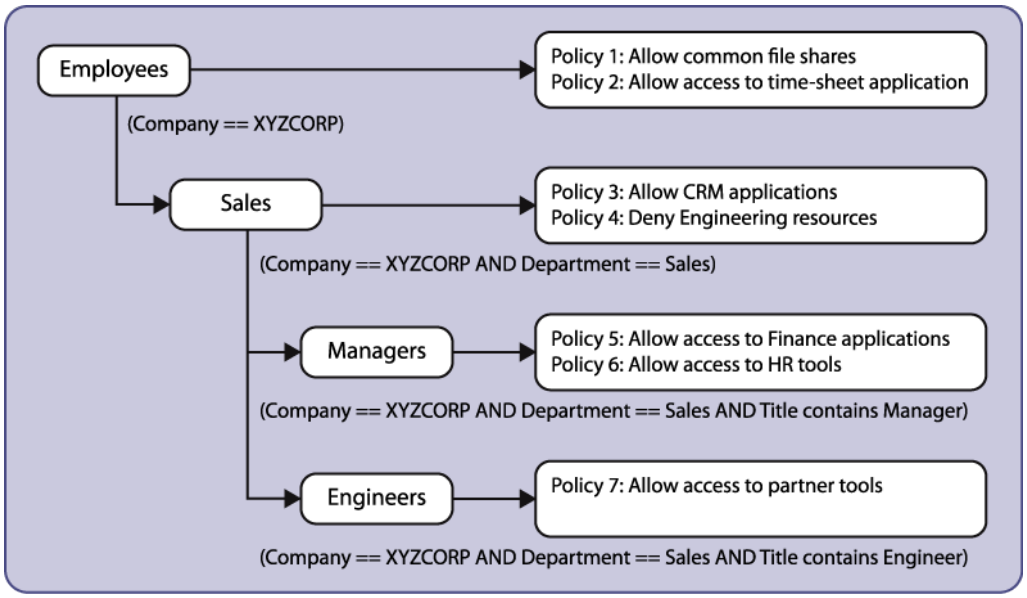


Figure 91: Hierarchical Role Management Example

Support for OR in Match Criteria

Prior to release 15.3, only AND was supported in the match criteria of user roles. From 15.3, OR is now supported also, so the user can have either AND or OR in the match criteria, but not both. For a particular role, the user can have all match criteria with AND, or have all the match criteria with OR. There is no restriction across roles with respect to role hierarchy and match criteria inheritance. The user can have the parent role with AND, and the child role with OR, or vice versa. The inheritance of match criteria to the child role from the parent uses AND as in previous releases.

```
Examples:
create identity-management role "EniBuildservers" match-criteria match-criteria "userName
contains
enibuild; OR ip-address == 10.120.89.0/24;
```

This example creates a role for enibuild servers, whose name contain enibuild or whose ip-addresses are in the range 10.120.89.1 - 10.120.89.255.

```
If the parent role has the match criteria as
"company == Extreme" AND "Title == Manager;"
And the child role has the match criteria as
"country == India;" OR "country == USA"
And the grandchild has the match criteria as
"department == Engineering"
```

All the managers who belong to Extreme Engineering, from both India and the USA, will be placed in the grandchild role.

Context Based Roles

Context based roles apply additional rules or policies to a port based on context related attributes for an identity. For example, consider a campus environment where a student logs into the network through a PC and also through a smart phone. Suppose that a role named Student already exists and applies basic policies for a student.
Also suppose that the administrator wants to apply additional policies for students accessing the network through smart phones.

To apply the additional policies, the administrator can create a role called Student_smartPhone as a child role of Student. The match criteria could include "title == Student; AND mac-oui == 00:1b:63:00:00:00/FF:FF:FF:00:00:00;", where the MAC address is the address of the smart phone. The additional policies can be added to the new child role. When the student logs in from the PC, the rules applicable to the Student role apply. When the student logs in from the smart phone, the policies for the Student_smartPhone role apply, in addition to those inherited from the Student role.

Note
A student logging on through a smart phone is placed under the Student_smartPhone role only if that role has a higher priority (lower numerical value) than the Student role.

Role Precedence and Priority

Note
When unknown MACs are identified from the FDB that do not have a role specified, they are marked "unauthenicate" in the IDM table. Whenever a port is restarted or FDB is flushed, these unauthenicate roles (identities) do not flush immediately and take less than a minute to flush out from the IDM table.

Roles are evaluated for identities in the following sequence:

1. The blacklist role is searched for the identity. If the identity is in the blacklist, the identity is denied access and role evaluation stops.
2. The whitelist role is searched for the identity. If the identity is in the whitelist, the identity is allowed access and role evaluation stops.
3. A local user-defined role is searched for the identity. If the identity is mapped to a local user-defined role, the identity is allowed access and role evaluation stops for all unknown/*LLDP* users. For Kerberos and network login users (except those authenticated through the local network login database), a query is sent to an LDAP server with the user attributes. If the Kerberos and network login users (except those authenticated through the local network login database) do not map to any local user-defined role , the identity is placed in authenticated role.

   Note
   The LDAP query can be disabled for specific types of network login users, and the LDAP query is disabled for locally authenticated network login identities.

4. When the switch receives LDAP attributes for an identity, the software evaluates the user-defined roles. If one or more user-defined roles match the identity attributes, and if those roles have a higher priority (lower numerical value) than the current role applied to the identity, the policies for the current role are removed and the policies for the user-defined role with the highest priority are applied.

   Note
   To support a change from the one role to another, the role priority for the new role must be higher than the current role.

5. Authenticated identities that cannot be placed in a user-defined role remain assigned to the authenticated role.
6. The unauthenticated role is applied to all identities that do not match any other roles.

Application of Rules or Policies

When the software makes the final determination of which default or user-configured role applies to the identity, the policies and rules configured for that role are applied to the port to which the identity connected. This feature supports up to eight policies and dynamic *ACL* rules per role (eight total).

When a dynamic ACL or policy is added to a role, it is immediately installed for all identities mapped to that role. Effective configuration of the dynamic ACLs and policies will ensure that intruders are avoided at the port of entry on the edge switch, thereby reducing noise in the network.

Note
The identity management feature supports wide key ACLs, which allow you to create many more match conditions for each ACL. For more information, see Wide Key ACLs on page 367.

The dynamic rules or policies that are installed for an identity, as determined by its role, are customized for that identity by inserting the MAC or IP address of the identity as the source address in the ACL rules. In ExtremeXOS release 12.5, identity manager inserted the IP address of the identity in all the ACL rules to be installed for that identity. Beginning with release 12.6, identity manager can insert either the MAC address or the IP address of the identity in all the ACL rules to be installed for that identity. By default, the MAC address of the identity is used to install the ACLs. Every network entity has a MAC address, but not all network devices have an IP address, so we recommend that you use the default configuration to install ACLs for network entities based on the source MAC address.

For additional information on creating ACLs, see ACLs on page 341. For additional information on creating policies, see Policy Manager on page 1448.

Role-Based Policy Enforcement

After user information is retrieved from the directory server, it is matched against a configured set of criteria and the user is then assigned to a specific role.

Pre-defined roles contain details of attributes with corresponding values to be used as match-criteria and the policies that need to be applied for that role. The administrator will be provided with a set of CLI commands to map association between role, match-criteria, and policies.

The following is a list of LDAP attributes that can be looked up in the LDAP server:

- Employee/User ID
- Title
- Department
- Company

- City
- State
- Country
- Email ID

Association Between Role and Attribute

Using CLI, various roles can be created with corresponding match criteria specified in attributes and values.

Association Between Role and Policy

When a policy is added to a role, the newly added policy will be applied to both existing users mapped to that role as well as new users who get mapped to this role in the future.

Match Criteria Inheritance

Beginning in release 15.2, a child role can inherit the match criteria of the parent role. The match criteria now does not need to be duplicated in all levels of the hierarchy.

Group Attributes Support

Network users can be mapped to a role based on group membership (distribution list) information. When a user is detected by identity manager, it retrieves the groups in which the detected user is member of from the LDAP server. Identity manager places the user under the appropriate role, based on group information and existing eight LDAP attributes.

You can specify the group name in the role's match criteria while creating the role. For example, the role creation command will appear as follows:

```
1  Create identity-management role Role1 match-criteria "memberOf==EXOSCLI-Review"
2  Create identity-management role Role2 match-criteria "title==Engineer; AND
memberOf==PI_SW"
```

A role's match criteria accepts all of the following operators: ==, !=, contains, AND, and OR.

Note
A combination of AND and OR is not supported in the match criteria definition of the role.

You can specify groups of the following types in match-criteria:

- direct-membership: the user is a direct member of the group specified in role match-criteria.
- hierarchical-membership: the user is not a direct member of the group specified, but comes under a specified group, per the hierarchy of the Active Directory (i.e., nested groups). Hierarchical groups are supported in Windows Server 2003 and later. Some LDAP servers may require special OID to perform a hierarchical search.

When a user is detected by identity manager, the following things occur:

- Identity manager retrieves eight LDAP attributes as supported before the 15.3 release, and also the Distinguished Name of the user.
- If any role's match criteria contains group attribute, a second LDAP query is created using the Distinguished Name of the user to retrieve all of the groups that the user is a member of. If an OID is configured for the hierarchical search, it will be used to form this LDAP query.
- Role determination takes place based on the group membership information and other LDAP attribute values.

The following optimizations are completed with respect to the LDAP query for Group Attributes:

- All of the group names used in every role configuration are collected and stored in a global database. When the LDAP query returns a list of the user's groups, the group names are cached against the user and used for role determination. The optimization is that only the group names used for role configuration are cached. The rest of the group names are discarded.
- The second LDAP query is not sent if the group attribute (i.e., memberOf) is not used in any role.

Network Zone Support for Policy Files

Network zone support helps users create a network zone and add multiple IP address and/or MAC addresses, which can then be used in the policy files.

With this new support, you can add a single attribute “source-zone” or “destination-zone” to an entry of a policy file. This entry is then expanded into multiple entries depending upon the number of IP and/or MAC addresses configured in that particular zone. If the zone is added to the policy with the keyword “source-zone,” the attributes that are configured in those particular zones, will be added as either source-address or ethernet-source-address in the policy, whereas, if the network-zone is added as destination-zone, the attributes will be added to the policies as destination-address or ethernet-destination-address.

Once you complete the changes in the zones, and issue a refresh of a specific zone, or all the zones, the policies that have corresponding zones configured as source-zone or destination-zone in their entries will be exapnded, and then refreshed in the hardware.

If you configure a policy such as the following to a port or *VLAN (Virtual LAN)* through applications such as IdMgr, *Extreme Network Virtualization (XNV)* or CLI:

```
Policy: test
entry e1 {
if match all {
    source-zone zone1 ;
}
Then {
    permit  ;
}
}
```

Upon refreshing the network-zone zone1, the policy will be expanded as below:

```
entry Cl0:0_10.1.1.1_E1 {
  if match all {
    source-address 10.1.1.1 / 32 ;
  } then {
    permit  ;
  }
}
entry Cl0:0_10.1.1.1_E2 {
  if match all {
    source-address 10.1.1.1 / 28 ;
  } then {
    permit  ;
  }
}

entry Cl0:0_12.1.1.0_E3 {
  if match all {
    source-address 12.1.1.0 / 24 ;
  } then {
    permit  ;
  }
}
```

This converted policy will be the one to be applied in the hardware.

Note

When the policy is configured in the network-zone, the zone may or may not have attributes configured with it. In cases where the network-zone does not have any attributes, the policy will be created with entries that do not have the network-zone attribute alone. For example, if you have a policy similar to the following:

```
Policy: test2
entry e1 {
if match all {
    source-zone zone2 ;
    protocol udp ;
}
then {
    permit  ;
}
}
entry e2 {
if match all {
    protocol tcp ;
}
then {
    permit  ;
}
}
```

and the network-zone “zone02” is created but not configured with any attributes, the policy would be as follows:

```
entry e2 {
protocol tcp;
```

```
}
then {
    permit;
}
}
```

Once the network-zone “zone2” is configured with one or more attributes and refreshed, the policy will be updated accordingly. Here the name of the entries that have source-zone or destination-zone will be changed. This is because each entry in the original policy that has a source-zone/destination-zone will be converted to a maximum of eight entries in the new policy.

A single policy can have one or more network-zones configured in it, and can also have the same network-zone in multiple entries with different combinations as other attributes are supported in the policy file. Similarly, the same network-zone can be configured to multiple policies. In cases where the policy has multiple network-zones, and only some of those network-zones are refreshed, the entries that correspond to those network-zones will be refreshed alone, and not the entries that correspond to the other network-zones.

Once a refresh of a network zone is issued, all the policies that have the specified network-zone will be modified, and a refresh for each of those policies will be sent to the hardware. The command will succeed only after getting a success return for all the policies that have this particular network-zone. If one of the policy’s refresh fails in the hardware, all of the policies that are supposed to refresh will revert to their previous successful state and the command will be rejected.

The configuration or refresh may fail if the attributes in the network zone are not compatible with the attributes in the corresponding entries of the policy. For example, in platforms that do not support wide-key or UDF, a policy entry cannot have Layer 2 attributes and Layer 4 attributes. In this case, if the entry has “protocol tcp,” and a network_zone that has an ethernet source address, the configuration will fail in the hardware.

Note

In the refresh failed case, the content of the policy and the content of the network-zone may go out of sync, as the policy will be reverted back to the last successful state, whereas the network zone will contain the last configured values.

For example, if we have the the network-zone configuration as follows:

```
create access-list network-zone zone1
configure access-list network-zone zone1 add ipaddress 10.1.1.1/32
configure access-list network-zone zone1 add ipaddress 10.1.1.1/28
```

and this is refreshed, and has been successfully installed in the hardware, the policy will look like this:

```
entry Cl0:0_10.1.1.1_E1 {
  if match all {
    source-address 10.1.1.1 / 32 ;
```

```
  } then {
    permit  ;
  }
}

entry Cl0:0_10.1.1.1_E2 {
  if match all {
    source-address 10.1.1.1 / 28 ;
  } then {
    permit  ;
  }
}
```

Now, if the user removes 10.1.1.1/28, and adds 10.1.1.1/24 to the network zone as below:

```
configure access-list network-zone zone1 delete ipaddress 10.1.1.1/28
configure access-list network-zone zone1 add ipaddress 12.1.1.0/24
```

and then does a refresh network-zone, and for some reason, the policy refresh fails, the policy and the network-zone will look as below:

```
entry Cl0:0_10.1.1.1_E1 {
  if match all {
    source-address 10.1.1.1 / 32 ;
  } then {
    permit  ;
  }
}

entry Cl0:0_10.1.1.1_E2 {
  if match all {
    source-address 10.1.1.1 / 28 ;
  } then {
    permit  ;
  }
}
create access-list network-zone zone1
configure access-list network-zone zone1 add ipaddress 10.1.1.1 255.255.255.255
configure access-list network-zone zone1 add ipaddress 12.1.1.0 255.255.255.0
```

Role Refresh

Role refresh allows you to enter a CLI command that triggers a reevaluation of role selection for one or all users. A role refresh can also trigger reevaluation of role selection for all users using a specific role.

After role evaluation completes for an identity, the role remains the same as long as the identity is present at the original location and no new high priority role matching this identity's attributes is created. Consider a situation where a Kerberos user is always present at a particular location. The switch detects traffic to and from the user periodically, so the user identity is never aged out. The user's role at this location remains the same as the role determined by identity manager when the user was detected at this location for the first time.

A network administrator might want to refresh a role for the following reasons:

- The user's LDAP attributes have changed. For example, the user's job title is changed from Engineer to Manager or his department is changed from Engineering to Marketing.
- The administrator has created a new role, which is more applicable to the user than his previous role. For example, the user was initially placed under the Engineer role because his department was Engineering, and now a new role called Test Engineer is a better match that considers both the user's department and title.

For both of the above situations, a role refresh triggers a role evaluation that would not otherwise occur as long as the user remains active at the current location. If the role refresh finds an LDAP user-defined role that matches the identity being refreshed, the identity manager queries the LDAP server to update the attributes provided by the LDAP server.

Switch Configuration Changes in Response to Identity Management Events

You can configure automatic switch configuration changes in response to identity management events. To do this, configure UPM profiles to respond to identity management events as described in Event Management System Triggers on page 2141.

Identity management events generate corresponding UPM events, including:

- IDENTITY-DETECT
- IDENTITY-UNDETECT
- IDENTITY-ROLE-ASSOCIATE
- IDENTITY-ROLE-DISSOCIATE

For instructions on displaying a complete identity management event list, see Event Management System Triggers on page 2141. The component name for identity management events is IdMgr.

Identity Management Feature Limitations

In the current release, the identity management feature has the following limitations:

- IPv4 support only. IPv6 to MAC bindings are not captured.
- For Kerberos snooping, clients must have a direct Layer 2 connection to the switch; that is, the connection must not cross a Layer 3 boundary. If the connection does cross a Layer 3 boundary, the gateway's MAC address gets associated with the identity.
- Kerberos snooping does not work on fragmented IPv4 packets.
- Kerberos identities are not detected when both server and client ports are added to identity management.
- Kerberos does not have a logout mechanism, so mapped identities are valid for the time period defined by the Kerberos aging timer or the Force aging timer.
- Kerberos snooping applied ACLs can conflict with other ACLs in the system.

IDM is not supported on LAG ports.

Configuring Identity Management

Note
When a switch is managed by NMS, it is possible to modify/delete configurations that are created by NMS. After editing via CLI, these changes are not reflected to NMS. Therefore, CLI configurations will be lost when NMS modifies the configuration further, because NMS configuration overrides CLI configuration. The following warning is printed so that users are aware that the CLI change is not permanent:

```
WARNING: An object that was created by Network Management System(NMS) has been
modified. The modification will not be reflected to the NMS and will not be
preserved if the object is subsequently modified by the NMS.
```

Note
The Identity Manager role-based *VLAN* feature will not be enabled on Netlogin enabled ports.

Basic Identity Management Feature Configuration

The following sections describe basic identity management configuration tasks.

Configuring the Maximum Database Size

- To configure the maximum size for the identity management database, use the following command:

  ```
  configure identity-management database memory-size Kbytes
  ```

- To set the maximum database size to the default value, use the following command:

  ```
  unconfigure identity-management database memory-size
  ```

Selecting the Access-List Source-Address Type

The identity management feature can install *ACL*s for identities based on the source MAC or source IP address. By default the MAC address of the identity is used to install the ACLs. Every network entity has a MAC address, but not all network devices have an

IP address, so we recommend that you use the default mac selection to install ACLs for network entities based on the source MAC address.

- To change the configuration for the access-list source-address type, use the following command:

  ```
  configure identity-management access-list source-address [mac | ip]
  ```

  Note

  You must disable identity management to change the current access-list source-address type configuration.

  By default, the identity's MAC address is used for applying the dynamic ACLs and policies. The dynamic ACLs or policies that are associated to roles should not have any source MAC address specified because the identity management feature will dynamically insert the identity's MAC address as the source MAC address. Similarly, if the ACL source address type is configured as ip, the dynamic ACLs or policies that are associated to roles should not have any source IP address specified.

Enabling and Disabling Identity Management

- To enable or disable the identity management feature, use the following admin-level commands.

  ```
  enable identity-management

  disable identity-management
  ```

  Identity manager does not detect and create identities for *FDB* blackhole and static entries.

  Note

  When the identity management feature is first enabled, the FDB entries previously learned on identity-management enabled ports are flushed.

Enabling and Disabling Identity Management on Ports

- To add or delete identity management on specific ports, use the following command:

  ```
  configure identity-management {add | delete} ports [port_list | all]
  ```

- To return to the default value, which removes all ports from the port list, use the following command:

  ```
  unconfigure identity-management ports
  ```

Enabling and Disabling SNMP Traps

- To enable the transmission of *SNMP (Simple Network Management Protocol)* traps for identity management low memory conditions, use the following command:

  ```
  enable snmp traps identity-management
  ```

- To disable SNMP traps for identity management, use the following command:

  ```
  disable snmp traps identity-management
  ```

Adjusting the Aging Time for Stale Entries

The stale-entry aging time defines when event entries in the identity management database become stale. To preserve memory, the software periodically uses a cleanup process to remove the stale entries.

- To adjust the period at which stale database entries are deleted (regardless of the database usage level), use the following command:

  ```
  configure identity-management stale-entry aging-time seconds
  ```

  Note
  For additional information on the stale-entry aging time and how it can be automatically adjusted by the software, see the command description for the above command.

- To set the stale-entry aging time to the default value, use the following command:

  ```
  unconfigure identity-management stale-entry aging-time
  ```

Resetting the Identity Management Configuration to the Default Values

To reset the identity management configuration to the default values, use the following command:

```
unconfigure identity-management
```

Adding and Deleting Entries in the Blacklist and Whitelist

To add or delete entries in the blacklist or whitelist, use the following commands:

```
configure identity-management blacklist add [mac mac_address {macmask} |
ip ip_address {netmask} | ipNetmask] | user user_name]

configure identity-management whitelist add [mac mac_address {macmask} |
ip ip_address {netmask} | ipNetmask] | user user_name]

configure identity-management blacklist delete [all | mac mac_address
{macmask} | ip ip_address {netmask} | ipNetmask] | user user_name]

configure identity-management whitelist delete [all | mac mac_address
{macmask} | ip ip_address {netmask} | ipNetmask] | user user_name]
```

Configuring Entries in Greylist

- To add or delete entries in greylist, use the following commands:

  ```
  configure identity-management greylist add user username

  configure identity-management greylist delete [all | user username]
  ```

- To display the entries in greylist, use the following command:

  ```
  show identity-management greylist
  ```

Configuring List-Precedence

- To configure or reset list-precedence, use the following commands:

  ```
  configure identity-management list-precedence listname1 listname2
  listname3

  unconfigure identity-management list-precedencee
  ```

- To display the list-precedence configuration, use the following command:

  ```
  show identity-management list-precedence
  ```

- To display the entries in the blacklist or whitelist, use the following commands:

  ```
  show identity-management blacklist
  show identity-management whitelist
  ```

Configuring Kerberos Snooping

Kerberos authentication or ticketing is used by Microsoft's Active Directory and by various Unix systems (including Linux and MAC OSX). The Kerberos snooping feature in the ExtremeXOS software collects identity information from Kerberos Version 5 traffic. This feature does not capture information from earlier versions of Kerberos.

Note
We recommend that you enable CPU DoS protect in combination with this feature to make sure the CPU is not flooded with mirrored Kerberos packets in the event of a DoS attack on Kerberos TCP/UDP ports. If the rate limiting capability is leveraged on capable platforms, it is applied on CPU mirrored packets.

Kerberos snooping is enabled when you enable identity management.

Note
Kerberos identities are not detected when both server and client ports are added to identity management.

Configuring a Kerberos Server List

By default, the identity management feature collects information from all Kerberos servers. However, this can subject the switch to DoS attacks targeted at Kerberos servers. To reduce the opportunities for DoS attacks, you can configure a Kerberos server list for identity management. When a Kerberos server list exists, identity management collects information only from the servers in the list.

- To add a server to the Kerberos server list, use the following command:

  ```
  configure identity-management kerberos snooping add server ip_address
  ```

- To delete a server from the Kerberos server list, use the following comma nd:

  ```
  configure identity-management kerberos snooping delete server
  [ip_address|all]
  ```

Note
Identity management supports configuration of up to 20 Kerberos servers.

Adjusting the Kerberos Snooping Aging Time

Kerberos does not provide any service for un-authentication or logout. Kerberos does provide a ticket lifetime, but that value is encrypted and cannot be detected during snooping. To enable the aging and removal of snooped Kerberos entries, this timer defines the maximum age for a snooped entry. When a MAC address with a corresponding Kerberos entry in identity manager is aged out, the Kerberos snooping aging timer starts. If the MAC address becomes active before the Kerberos snooping aging timer expires, the timer is reset and the Kerberos entry remains active. If the MAC address is inactive when the Kerberos snooping aging timer expires, the Kerberos entry is removed.

- To configure the Kerberos snooping aging time, use the following command:
  `configure identity-management kerberos snooping aging time minutes`
- To reset the Kerberos snooping aging time to the default value, use the following command:
  `unconfigure identity-management kerberos snooping {aging time}`

Note
The default value for this command is `none`, which means that an identity discovered through Kerberos snooping is removed immediately on the aging out of the identity MAC address by the *FDB* manager.

Forced Kerberos Logout

To force the removal of all identities discovered through Kerberos snooping, use the following command:

`configure identity-management kerberos snooping force-aging time [none | minutes]`

Configure Default and User-Defined Roles

Creating and Deleting User-Defined Roles

- To create or delete a role, use the following commands:
  `create identity-management role role_name match-criteria match_criteria {priority pri_value}`
  `delete identity-management role {role-name | all}`
- To create or delete a child role, use the following commands:
  `configure identity-management role role_name add child-role child_role`
  `configure identity-management role role_name delete child-role [child_role | all]`

Configuring Rules or Policies for Default and User-Defined Roles

The default authenticated and unauthenticated roles contain no rules or policies. When you first create a user-define role, it also contains no rules or policies.

To add or delete a rule or policy from a role, use the following commands:

```
configure identity-management role role_name [add dynamic-rule rule_name
{ first | last | { [before | after] ref_rule_name}}]

configure identity-management role role_name add policy policy-name
{first | last {[before | after] ref_policy_name}}

configure identity-management role role_name delete dynamic-rule
[rule_name | all]

configure identity-management role role_name delete policy [policy-name
| all]
```

Configuring LDAP Server Access

- To add or remove LDAP server connections for retrieving identity attributes, use the following commands:

  ```
  configure ldap {domain domain_name} add server [host_ipaddr |
  host_name] {server_port} {client-ip client_ipaddr} {vr vr_name}
  {encrypted sasl digest-md5}
  ```

- To create a new domain, use the following command:

  ```
  create ldap domain domain_name {default}
  ```

- To configure credentials for accessing an LDAP server, use the following command:

  ```
  configure ldap {domain [domain_name|all]} bind-user [user_name
  {encrypted} password | anonymous]
  ```

- To specify a base domain name to be added to usernames in LDAP queries, use the following command:

  ```
  configure ldap {domain [domain_name|all]} base-dn [base_dn | none |
  default]
  ```

- To specify a domain as default, use the following command:

  ```
  configure ldap domain domain_name [default | non-default]
  ```

- To enable or disable LDAP queries for specific network login types, use the following command:

  ```
  configure ldap { domain [ domain_name | all ] } [enable|disable]
  netlogin [dot1x | mac | web-based]
  ```

- To configure bind-user for LDAP queries, use the following command:

  ```
  configure ldap {domain [domain_name|all]} bind-user [user_name
  {encrypted} password | anonymous] {domain [domain_name|all]}
  ```

- To delete an LDAP server, use the following commands:

  ```
  configure ldap {domain [domain_name|all]} delete server [host_ipaddr |
  host_name] {server_port} {vr vr_name}
  ```

- To delete a domain, use the following command:

  ```
  delete ldap domain [domain_name | all]
  ```

Support for Multiple Windows Domains

Some organizations are large enough to use multiple Windows domains (sub-domains) in their networks. Each Windows domain can have its own LDAP server.

In previous releases, identity manager supported up to eight LDAP servers which are assumed to be replicas on the same domain (default base-dn). From the 15.2 release, identity manager supports multiple Windows domains.

LDAP Servers in Different Domains

In 15.2, identity manager can service users under different domains. You can configure different domains and add different LDAP servers for these different domains. When adding an LDAP server to identity manager, you can specify the domain under which the server is to be added.

- You can configure a base-dn and a bind user for each domain.
- Base-dn is assumed to be the same as the domain name unless explicitly configured otherwise. (Base-dn is the LDAP directory under which the users are to be searched.)
- For users upgrading from older configurations, the base-dn configured on an older version now becomes the default domain name. This can be changed later if required.
- For users upgrading from older configurations, the LDAP servers configured on older versions are now servers under the default domain.
- You can now add up to eight LDAP servers to each of the user-configured domains.

LDAP Connections

Identity manager tries to maintain LDAP connections with one of the servers in each of the configured domains. LDAP queries for users logging on to those domains will be sent to the respective servers or to a server on the default domain if the user does not fall under any configured domain. The LDAP server might choose to close the connection after a timeout.

LDAP Process

Identity manager tries to bind to one of the configured LDAP servers in each of the user-configured domains.

When a new user is detected, the user's domain is used to determine the LDAP server to be contacted for the user's details.

If there is a match, the LDAP server corresponding to that domain is chosen and the LDAP search request for the user attributes is sent to that LDAP server.

If the domain does not match any of the configured domains, LDAP query is sent to a server in the default domain.

Changing the Role Priority

The role priority is defined when a role is created. To change the priority for a role, use the following commands:

```
configure identity-management role role_name priority pri_value
```

Configuring Kerberos Authentication Type

To configure Kerberos authentication type:

```
create xml-notification target NetSight url
https://172.29.170.100:8443/fusion_jboss/XosIDM vr VR-Default

configure xml-notification target NetSight user root encrypted-auth
cm9vdDpsNGIwbDRiMCE=

configure xml-notification target NetSight add idMgr

enable xml-notification NetSight

configure log filter "XML_TEST" add events IdMgr
```

To send XML events:

```
create xml-notification target netsight url https://10.127.3.183:8443/axis/services/event
vr VR-Mgmt
 configure xml-notification target netsight user Administrator encrypted-auth
QWRtaW5pc3RyYXRvcjpFeHRyZW1lXzIx

 create xml-notification target netsight_new url https://10.127.3.183:8443/fusion_jboss/
XosIDM vr VR-Mgmt
 configure xml-notification target netsight_new user Administrator encrypted-auth
QWRtaW5pc3RyYXRvcjo=eHRyZW1lXzIx

 create xml-notification target test url http://10.127.3.183:8080/axis/services/event vr
VR-Mgmt
 configure xml-notification target test user Administrator encrypted-auth
QWRtaW5pc3RyYXRvcjpFeHRyZW1lXzIx

 enable xml-notification test

 create xml-notification target new url http://10.127.3.183:8080/fusion_jboss/XosIDM vr
VR-Mgmt
 configure xml-notification target new user Administrator encrypted-auth
QWRtaW5pc3RyYXRvcjpFeHRyZW1lXzIx
 enable xml-notification new
 configure xml-notification target netsight add idMgr
 configure xml-notification target netsight_new add idMgr
 configure xml-notification target test add idMgr
 configure xml-notification target new add idMgr
```

Managing the Identity Management Feature

Clearing the Identity Management Counters

To clear the statistics counters for the identity management feature, use the following commands:

```
clear counters

clear counters identity-management
```

Refreshing the Role Selection for Users

To refresh role evaluation for a specified user, for all users, or for all users currently using the specified role, use the following commands:

```
refresh identity-management role user [user_name {domain domain_name} |
all {role role_name}]
```

Enabling/Disabling Snooping Identities

The identity management feature makes the edge of the network more intelligent by providing access to the devices/users in the network. The identity manager detects the identities through the following protocols:

- *FDB*
- IPARP
- IPSecurity *DHCP* Snooping
- *LLDP*
- Netlogin
- Kerberos

By default, identity management detects identities through all of the above mentioned protocols. There is no way for the administrator to disable the detection of the identities tht are triggered through the above protocols.

This feature now provides the administrator an option to enable/disable the detection of the identities that are triggered through any of the above protocols. The administrator can now control the identity detection through any of the protocol triggers at port level. This configuration can be applied to identity management-enabled ports only. An error is received if this configuration is applied to identity management-disabled ports.

As part of this feature, the limitation of FDB entries getting cleared on enabling identity management on a port is removed. The identity mangement module will retrieve the FDB entries learned on the identity management-enabled ports and create the identity accordingly.

Note

All types of Netlogin identity will not be detected if the netlogin detection is disabled.

Enabling Kerberos identity detection does not create identities for previously authenticated clients.

Displaying Identity Management Information

Displaying Database Entries

Log in as an admin-level user. To display the entries in the identity management database, enter the following command:

```
show identity-management entries {user id_name} {domain domain} {ports
port_list} {mac mac_address} {vlan vlan_name} {ipaddress ip_address}
{detail}
```

Displaying Configuration Information

Use the following command to display the current configuration of the identity management feature:

```
show identity-management
```

```
show identity-management role {role-name} {detail}
```

Displaying Statistics

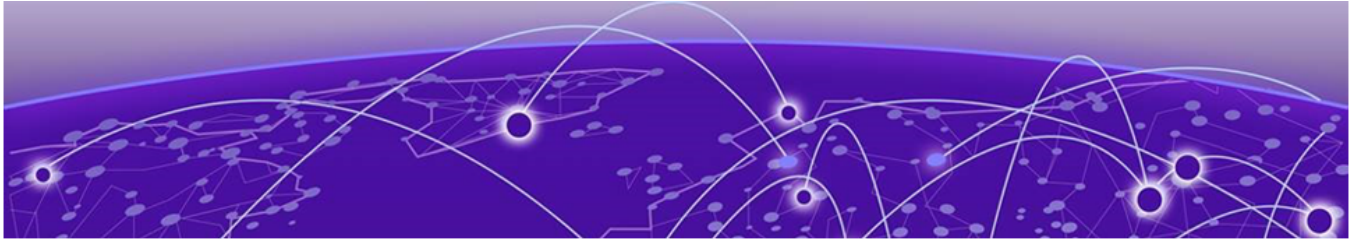
- To display operating statistics for the identity management feature, use the following command:

  ```
  show identity-management statistics
  ```

- To clear the statistics counters for the identity management feature, enter either of the following commands:

  ```
  clear counters
  ```

  ```
  clear counters identity-management
  ```



Integrated Application Hosting

Integrated Application Hosting Overview on page 868
Integrated Application Hosting (IAH) Supported Platforms on page 870
Integrated Application Hosting (IAH) Limitations on page 870
Creating an Ubuntu Virtual Machine on page 870
Configuring Integrated Application Hosting (IAH) Virtual Machines on page 871
Integrated Application Hosting (IAH) Commands on page 873
Integrated Application Hosting Use Cases on page 875

This feature is not supported in this release.

Integrated Application Hosting Overview

The Integrated Application Hosting (IAH) feature provides the ability to host guest VMs on ExtremeXOS-based switches.

The IAH infrastructure includes:

- ExtremeXOS Virtualization Manager component—responsible for guest VM management and control, management of networking connectivity, management of compute resources, and complete isolation of activity from ExtremeXOS.
- KVM (Kernel Virtual Machine)—Linux kernel feature enabling hardware-accelerated virtualization.
- QEMU (Quick EMUlator)—an open-source emulator and hypervisor.

VMs can be packaged as either OVAs (Open Virtual Appliance) or qcow2 or any QEMU-compatible (including VMDK) disk images.

Requirements

IAH requires the following:

- Premier or Extreme Platform ONE license (for more information about licenses, see the Switch Engine v33.6.1 Licensing Guide)
- Solid State Storage Device SSD-120 (see SSD-120 Device on page 869).

SSD-120 Device

The SSD-120 device is a dedicated storage device, required for the Integrated Application Hosting (IAH) feature, and provides support for virtual machines (VMs) on ExtremeXOS-based switches, including:

- Storage of VM installation images (OVA and downloaded qcow2 or any QEMU-compatible (including VMDK) disk images)
- VM configuration files (libvirt XML format)
- Disk images for installed VMs
- Libvirt configuration files

Third-Party Virtual Machine (TPVM)

SSD-120 contains Third-Party Virtual Machine (TPVM), which is an Ubuntu-based VM appliance supporting a number of network-centric applications. TPVM allows you to run applications such as Docker Container, Syslog server, SNMP server, and RESTful applications, among others. TPVM runs as a separate, independent VM, sharing the host CPU, RAM, hard disk drive, and management resources with ExtremeXOS.

Folder Structure

The folder structure of SSD-120 is available at `/usr/local/vm`. The `/usr/local/vm/packages` folder is for storing downloaded OVA, qcow2, and QEMU-compatible VM installation files.

Virtual Machine (VM) Network Connectivity

Running virtual machines (VMs) can access the Management VLAN when an associated VM port is assigned an IP address on the Management VLAN. This access is intended for management/maintenance purposes, and shares the bandwidth of the management port with switch management.

The normal connectivity between a guest VM and the network is by dedicated ports. Any management operations (CLI commands, SNMP gets/sets, API calls, etc.) that can be applied to a front-panel port can also be applied to a dedicated port. Only one guest VM can be configured per dedicated port.

To configure Integrated Application Hosting (IAH) VMs, including adding dedicated ports, see Configuring Integrated Application Hosting (IAH) Virtual Machines on page 871 .

Integrated Application Hosting (IAH) GUI Support

The video output device of a guest OS is accessible using any VNC client.

For information about configuring VNC, see Configuring Integrated Application Hosting (IAH) Virtual Machines on page 871 and Integrated Application Hosting (IAH) Commands on page 873. For information about VNC limitations, see Integrated Application Hosting (IAH) Limitations on page 870.

Integrated Application Hosting (IAH) Supported Platforms

Integrated Application Hosting (IAH) is supported on ExtremeSwitching 5720, Extreme 7520, and Extreme 7720 series switches running a Premier or Extreme Platform ONE license. For more information about licenses, see Switch Engine v33.6.1 Licensing Guide .

Integrated Application Hosting (IAH) Limitations

Integrated Application Hosting (IAH) has the following limitations:

- A maximum of two CPU cores can be dedicated to VMs.
- Multiple guest VMs per dedicated port is not supported.
- On stacks, the VMs do not migrate to backup with a failover. You will need to restore the original master as master to access your VMs again.
- Only OVA, qcow2, and any QEMU-compatible files are supported. Compatibility issues might occur when using third-party OVA files. The image formats VMDK and qcow2 are generally more reliable than third-party OVA files.
- You cannot rename a VM after creating it.
- VNC servers are only accessible using SSH tunnel for security reasons.
- VNC support is not available on non-primary stack nodes.

Creating an Ubuntu Virtual Machine

To create an Ubuntu virtual machine (VM) qcow2 disk image:

1. Download Ubuntu server 16.04 LTS or 18.04 LTS:
   - http://releases.ubuntu.com/18.04.2/ubuntu-18.04.2-live-server-amd64.iso
   - http://releases.ubuntu.com/16.04.6/ubuntu-16.04.6-server-amd64.iso
2. Use virt-manager on a Linux system to create a VM with 2 CPUs, 4 GB RAM and 32 GB hard disk of qcow2 format. Select the option `customize configuration before install`.
3. Customize the configuration to change the hard disk bus type to `VirtIO`.
4. Start the VM.
5. Follow the prompts to complete the Ubuntu installation.
6. Enable the serial console:
   a. Edit the file `/etc/default/grub` to have the following settings:

      ```
      GRUB_CMDLINE_LINUX="console=tty1 console=ttyS0,115200"
      GRUB_TERMINAL="console serial"
      GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --stop=1"
      ```

   b. Update `/boot/grub/grub.cfg`: `update-grub`.

7. (Optional) Update the system and install additional software:
   a. Enable all repositories for apt:
      - apt-add-repository restricted
      - apt-add-repository universe
      - apt-add-repository multiverse
      - apt update
   b. Upgrade Ubuntu and software: `apt upgrade`.
   c. Reboot if required.
   d. Install additional software:
      - apt install build-essential
      - apt install checkinstall
      - apt install iperf apt install mtools
      - apt install netperf
      - apt install qemu-guest-agent
      - apt install tshark
      - apt install valgrind
      - apt install vim-gnome
      - apt install wireshark
      - apt install xterm
8. Fill the unused disk space with zeros so that the image interacts well with compression: `dd if=/dev/zero of=/mytempfile` (will eventually error out with 'out of space', but may take a while):

   ```
   rm -f /mytempfile
   ```

9. Shutdown the VM.
10. Copy the qcow2 disk image onto an ExtremeXOSswitch and install it as a VM. The image is located at `/var/lib/libvirt/images/`.

Note
If you do not have root access, you the following command to create a copy of the disk image: `virsh vol-download --pool default disk_name.qcow2 my_copy.qcow2`.

Note
The image size can be reduced substantially because of all the adjacent zeroes: `qemu-img convert -O qcow2 my_copy.qcow2 my_reduced_copy.qcow2`.

Configuring Integrated Application Hosting (IAH) Virtual Machines

Ensure that you have the correct license installed on the switch (for more information about licenses, see the Switch Engine v33.6.1 Licensing Guide), and that you have purchased a Solid State Storage Device (SSD) SSD-120.

To set up an Integrated Application Hosting (IAH) guest virtual machine (VM):

1. Install the Solid State Storage Device (SSD) SSD-120:
   a. Power down the switch.
   b. On the rear panel of the switch, remove the cover from the SSD slot.
   c. Insert SSD-120 into the SSD slot.
   d. Power up the switch.
2. Download a VM installation file (OVA, qcow2, or any QEMU-compatible file) to SSD-120 to `/usr/local/vm/packages` using the command `scp2 {cipher cipher} {mac mac} {compression [on | off]} {port portnum} {vr vr_name} user [hostname | ipaddress]:remote_file local_file` . Alternatively, you can use the Third-Party Virtual Machine (TPVM), which is included on SSD-120.
3. Install the VM using one of the following commands:
   - Disk image: `create vm vm_name image image_file {memory memory_size} {cpus num_cpus} {slot slot_ID}{vnc [none | vnc_display]}`
   - OVA file: `create vm vm_name ova ova_file {memory memory_size} {cpus num_cpus} {slot slot_ID} {vnc [none | vnc_display]}`

     Note
     Compatibility issues might occur when using third-party OVA files. The image format qcow2 is generally more reliable.

   The default for number of CPUs allocated is 1. The default amount of RAM allocated to a VM is 4,096 MB. You can change this with the `create vm` command as needed.

   The VNC server only listens to the switch's loopback IP address (127.0.0.1). You need to forward TCP traffic from the server's port on the switch to a port that the client software is running on using an *SSH (Secure Shell)* tunnel. On the switch, VNC port numbers range from 5,900 to 5,915.

   To create this setup, use the following command:

   ```
   ssh admin@mgmt_ip -L port_on_client_system:127.0.0.1:port_on_switch
   ```
4. Add IAH dedicated ports to the VM.

   ```
   configure vm vm_name {add | delete} ports portlist
   ```
5. Choose whether or not the VM should start up automatically when ExtremeXOS starts. Automatic start-up is disabled by default. To enable automatic start-up, use the following command:

   ```
   enable vm vm_name autostart
   ```

6. To open a session to the VM's serial console, use the following command:

   ```
   open vm vm_name {console}
   ```

   Note
   TPVM is configured to use a serial console.

   Note
   You cannot access the serial console before starting a VM. You must start the VM, and then reboot it to gain serial console access.

7. Optionally, add virtual interfaces to the VM using the following command (maximum 16 virtual interfaces):

   ```
   configure vm vm_name add virtual-interface port port {vlan vlan_id}
   {name vf_name}
   ```

8. Optionally, change the vm's disk bus or controller, using the following command:

   ```
   configure vm vm_name disk bus-type bus_type
   ```

   The default bus type is VirtIO, but some operating systems are do not support this, and as a consequence, the VM will fail to boot. You can configure the bus type to IDE or SCSI.

9. Start the VM.

   ```
   start vm vm_name
   ```

To check the status of the VM, use the following commands:

- For operational status information: `show vm {vm_name | detail}`
- For interfaces information: `show vm vm_name guest interfaces`

If needed to save disk space, after successful creation of the VM, you can delete the VM package file at `/usr/local/vm/packages`.

For additional IAH commands, see Integrated Application Hosting (IAH) Commands on page 873.

Integrated Application Hosting (IAH) Commands

The following topics shows all Integrated Application Hosting (IAH) for guest virtual machines (VMs) commands.

Creating VMs

```
create vm vm_name ova ova_file {memory memory_size} {cpus num_cpus}
{slot slot_ID} {vnc [none | vnc_display]}
```

```
create vm vm_name image image_file {memory memory_size} {cpus num_cpus}
{slot slot_ID}{vnc [none | vnc_display]}
```

Configuring VMs

```
configure vm vm_name {add | delete} ports portlist

open vm vm_name {console}

configure vm vm_name cpus num_cpus

configure vm vm_name memory memory_size

configure vm vm_name add virtual-interface port port {vlan vlan_id}
{name vf_name}

configure vm vm_name delete virtual-interface [name vf_name | mac
mac_addr]

configure vm vm_name vnc [none | vnc_display]

configure vm vm_name disk bus-type bus_type
```

Starting/Stopping/Suspending/Resuming VMs

```
start vm vm_name

stop vm vm_name [forceful | graceful]

restart vm vm_name {forceful | graceful}

suspend vm vm_name

resume vm vm_name
```

Enabling/Disabling Automatic Start-up for VMs

```
enable vm vm_name autostart

disable vm vm_name autostart
```

Deleting VMs

```
delete vm vm_name
```

Showing VM Information

```
show vm {vm_name | detail}

show vm vm_name guest interfaces

show vm vm_name virtual-interface {vf_name}
```

Clearing VM Storage

```
clear vm storage
```

Saving VMs

```
save vm vm_name state
```

Removing VMs

To remove all installed VMs, use the following command with the all option:

```
unconfigure switch {all | erase [all | nvram]}
```

Note
VM installation files (OVA, qcow2, QEMU-compatible) are not removed with this command.

Be aware that using this command generally resets the switch back to factory defaults, which includes removing the configuration file, management IP address, failsafe account, and SummitStack-specific parameters, and the rebooting the switch.

Exporting VM Disk Images

```
save vm vm_name image image_file
```

Integrated Application Hosting Use Cases

The following section provides use cases for the Integrated Application Hosting (IAH) feature.

Palo Alto Firewall Virtual Machine (VM) Use Case

Palo Alto Networks offers a virtualized version of its firewall that is compatible with the X86_64-based compute complex provided by the Extreme Networks Integrated Application Hosting (IAH) system. The following sections detail the procedure for integrating this VM into IAH.

Solution Requirements and Design

The virtual machine will load, start, and function. The first port configured to the VM is assigned to the Palo Alto management interface. Subsequent ports attach sequentially from Ethernet 1/1. Sub-interfaces do work with this VM, so it is easily possible to deploy the firewall with a single Integrated Application Hosting (IAH) forwarding plane

port, and break out ports using VLANs on the switch. The serial terminal function of Switch Engine works with no extra configuration required.

Table 47: Services Offered by the IAH Compute Environment on Switch Engine

Switch Models	Number of CPUs	Available RAM	Available Storage	Dedicated Management Port	Management Port BW	Number of Sideband Ports	Sideband BW Supported
5720-24MXW	2	4GB	120GB	Yes	1G	1	10G
5720-48MXW	2	4GB	120GB	Yes	1G	1	10G
7520-48Y	8	16GB	120GB	Yes	1G	1	10G
7520-48XT	8	16GB	120GB	Yes	1G	1	10G
7720-32C	8	16GB	120GB	Yes	1G	2	10G

Table 48: Extreme Integration Elements

Minimum Switch Engine Version	Switch Model	Resources Required from IAH to Support Solution	License Requirement
32.3	5720 (MXW models)	2 CPU cores 5,730 KB of RAM	Premier or Extreme Platform ONE license
32.4	7520 (all models)	8 CPU cores 5,730 KB of RAM	Premier or Extreme Platform ONE license
32.5	7720	8 CPU cores 5,730 KB of RAM	Premier or Extreme Platform ONE license

Table 49: Palo Alto Integration Elements

AP OS	Application	Resources Required from IAH to Support Solution	License Requirement
9.1.2	PAN VM-100 (PA-KVM-9.1.2.qcow2)	2 CPU cores 5,730 KB of RAM	Standard Subscription
9.1.2	PAN VM-100 (PA-KVM-9.1.2.qcow2)	2 CPU cores 8,192 KB of RAM	Standard Subscription

Note

Virtual Interface mapping is not visible to the Palo Alto Firewall virtual machine. If all mapped interfaces are virtual, the VM will panic and go into maintenance mode.

Installation and Configuration Procedure

For this procedure port 57 is assigned to the “inside” VLAN on the switch. This VLAN requires an internal DHCP server on the 192.168.1.x/24 subnet. To create this, use the following command:

```
# configure Default add ports 57
```

To verify the Default VLAN configuration:

```
# show inside
5720-42MW # sh inside
VLAN Interface with name inside created by user
    Admin State:         Enabled     Tagging:   802.1Q Tag 10
    Description:         None
    Virtual router:      VR-Default
    IP Anycast:          Disabled
    IPv4 Forwarding:     Disabled
    IPv4 MC Forwarding:  Disabled
    Primary IP:          192.168.1.254/24
    IPv6 Forwarding:     Disabled
    IPv6 MC Forwarding:  Disabled
    IPv6:                None
    STPD:                None
    Protocol:            Match all unfiltered protocols
    Loopback:            Disabled
    NetLogin:            Disabled
    QosProfile:          None configured
    Egress Rate Limit Designated Port: None configured
    Flood Rate Limit QosProfile:       None configured
    Suppress ARP:        Disabled
    Suppress ND:         Disabled
    Proxy ARP:           Entry required
    Ports:   2.           (Number of active ports=1)
       Untag:       1,*57(Insight)
```

1. Transfer the firewall image to the switch by entering the directory of the images:

   ```
   # cd /usr/local/vm/packages/
   ```

   Note
   If you are unable to change to this directory, you are likely missing a Core license on your switch.

2. Transfer the image to the server/host with SFTP/SCP services:

   ```
   # scp2 vr VR-Default <username>@<IP Address of SCP Host>:PA-KVM-9.1.2.qcow2 PA-
   KVM-9.1.2.qcow2
   ```

3. Create the VM:

   ```
   # create vm PAN image PA-VM-KVM-9.1.2.qcow2 cpu 2 memory 5730
   Creating
   VM.....................................................................................
   ......................
   VM PAN created successfully.
   ```

4. Configure the VM with interfaces.

   Assign two interfaces to the firewall: port 57 and 58. The VM picks up the first interface that comes up. The management port is shared with the front panel management port of the switch and should be addressed for that management interface. Port 57 on the ExtremeSwitching X465 is directly connected to the forwarding plane of the switch.

   ```
   # configure vm PAN add ports mgmt
   # configure vm PAN add ports 57
   # configure vm PAN add ports 58
   ```

   Adding these ports in sequence maps them in sequence as the VM boots. Palo Alto takes the first port for management, then the second port as Ethernet 1/1, and the third as Ethernet 1/2. The ports are configured on the switch to VLAN Mgmt (10.10.10.x), VLAN inside (192.168.1.x), and VLAN outside (10.10.100.x). Each VLAN has a DHCP server available for their respective subnets. If you want the VM to start automatically every time the switch is booted, add this command:

   ```
   # enable vm pan autostart
   ```

5. Verify the configuration of the VM:

   ```
   # show vm pan
       VM Name:         pan
       State:       Stopped
       Memory size: 5370 MB
       CPUs:        2
       Auto-start:  Disabled
       VNC:         127.0.0.1:1 (Port 5901)
       Disk:        vda
           Source:                  /mnt/vmdisk/.vm/pan_PA-VM-KVM-9.1.2.qcow2
           Disk bus type:           virtio
           Allocated size in bytes: 64424509440 (60.00 GB)
           Physical size in bytes:  5583732736 (5.20 GB)
           Read requests:           0
           Bytes read:              0
           Write requests:          0
           Bytes written:           0
       Network interfaces:
           Attached switch ports:   mgmt,57-58
   ```

6. Start the VM:

   ```
   # start vm pan
   ```

7. Open a console connection into the VM:

   ```
   # open vm PAN
   Connected to domain PAN
   Escape character is ^Y
   ```

   Note
   The Palo Alto firewall VM takes a moment to boot-up. It may be a few minutes before the prompt appears on the CLI.

   The firewall will report the IP address it retrieved in the CLI before logon: DHCP: new IP 10.10.10.149 : mask 255.255.255.0

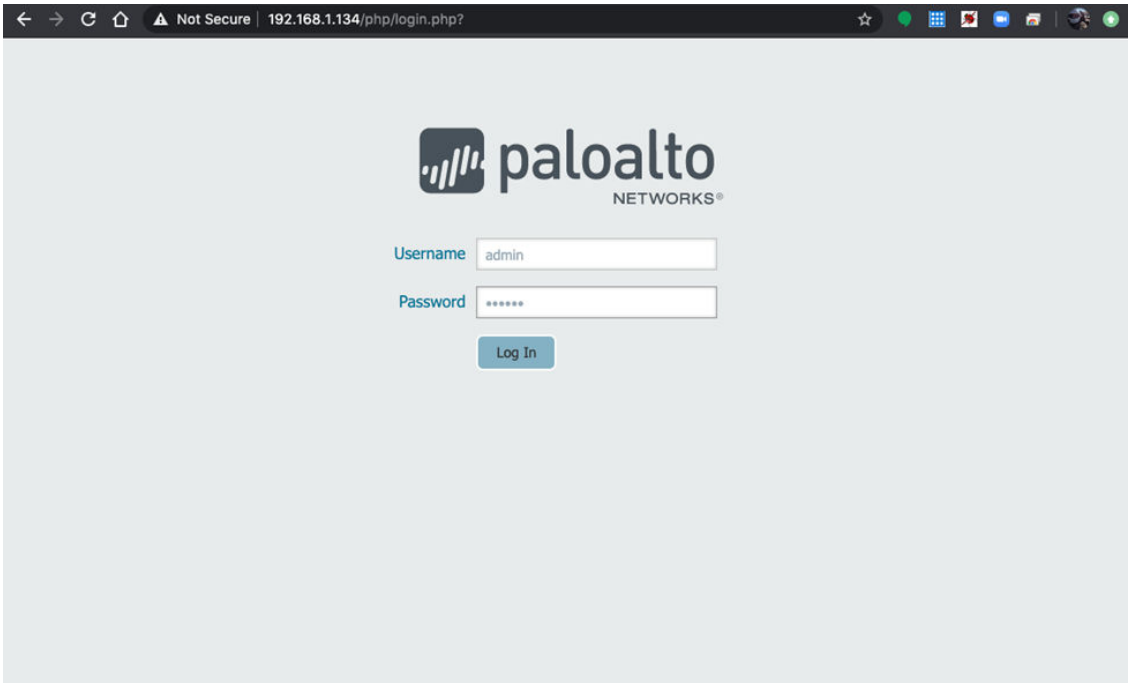


Figure 92: Palo Alto Firewall Logon Window

8. Open the serial connection to the firewall VM on the ExtremeXOS CLI:

```
Connected to the domain pan
Escape character is ^Y

PA-VIM Login:
```

9. Log on with the credentials: admin/admin.
10. Connect using VNC:

    The IAH complex on ExtremeXOS also offers a proxy to a switch with a VNC connection to the VM. You can map up to 16 connections. In this case, VNC connection 1 is mapped. Opening this VNC session provides direct access to the VM “screen”. No special configuration of the virtual machine is required.

    You can see this configuration using the command `show vm pan`. The mapped port is 5901, but it is not directly accessible. To reach this port, it is necessary to map an SSH tunnel on the client accessing the switch. On MacOS and Linux environments, us the following command:

```
# ssh admin@<ip address of the switch> -L 5901:127.0.0.1:5901
```

    You are prompted for your password, and a successful logon to the switch also creates an SSH tunnel to the VNC server. Use your favorite VNC client to open: 127.0.0.1:1.
11. Activate the firewall interface and assign VLANs for zones:

    Configure according to Palo Alto instructions. The firewall interface is Eth1/1 and Eth1/2. This example is configured with each in Layer 3 mode. The interfaces are directed to request IP addressing from the switch DHCP server.

12. Confirm the DHCP server setup:

```
# show dhcp-server
    VLAN "inside":
    DHCP Address Range   : 192.168.1.100->192.168.1.110
    Netlogin Lease Timer : Not configured (Default = 10 seconds)
    DHCP Lease Timer     : Not configured (Default = 7200 seconds)
    Default Gateway      : 192.168.1.1
    Primary DNS Server   : 8.8.8.8
    Ports DHCP Enabled   : 1,57

    ================================================================================
    IP                MAC                State      Lease Time Left
    ================================================================================
      192.168.1.100   48:9b:d5:eb:1e:42   Assigned   0001:22:17

VLAN "outside":
    DHCP Address Range   : 10.10.100.100->10.10.100.110
    Netlogin Lease Timer : Not configured (Default = 10 seconds)
    DHCP Lease Timer     : Not configured (Default = 7200 seconds)
    Ports DHCP Enabled   : 58

    ================================================================================
    IP                MAC                State      Lease Time Left
    ================================================================================
      10.10.100.100   48:9b:d5:eb:1e:43   Assigned   0001:22:17
```

13. Continue configuring the firewall. For more information, see the Palo Alto Networks documentation.

    The 5720 series is configured for two firewall interfaces mapped to non-trunked ports on the switch in Layer 3 mode for a total of two zones: inside and outside. DHCP was used to assign IP addresses. You can chose instead to use tagged VLANs on the switch ports 57 and 58 to present more switch VLANs to the firewall by sub-interfaces or VLANs. Using two sideband ports provides a theoretical bandwidth of 20Gb/s in and 20Gb/s out for firewall inspection.

14. Configure the interfaces to allow PING and other services, so that bi-directional communication can be verified.

    Network > **Network Profiles** > **Interface Management**.

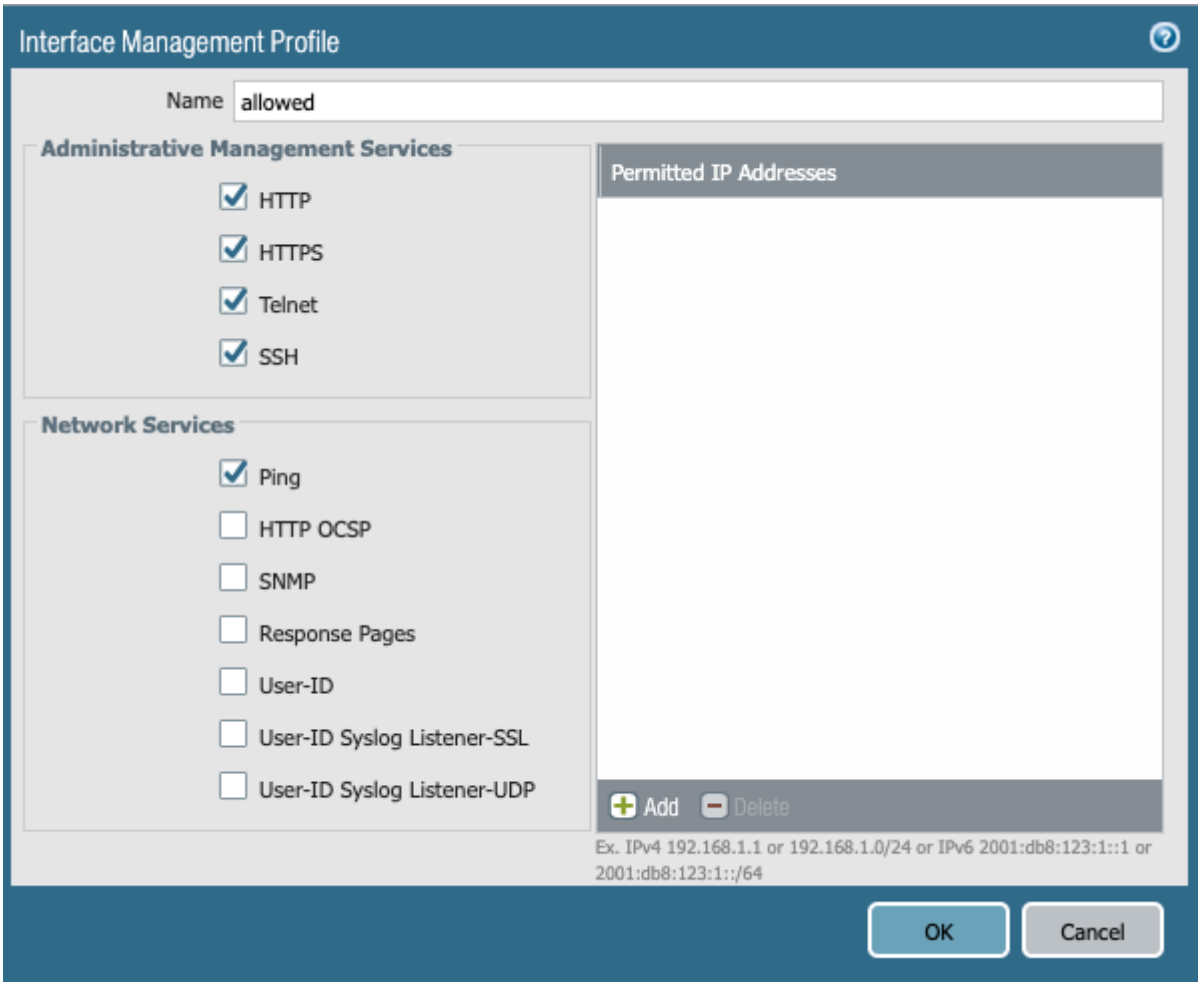


Figure 93: Palo Alto Firewall Interface Management Profile Window

15. PING your interfaces and populate the IPARP and FDB tables on the switch.
16. Check the interface mapping against the switch MAC addresses. From the VM side, use the command:

```
admin@PA-VM> debug show vm-series interfaces all

Interface_name    Base-OS_port     Base-OS_MAC           PCI-ID       Driver
  Mgt               eth0       52:54:00:7d:e5:c1                    virtio_net
 Ethernet1/1         eth1       48:9b:d5:eb:1e:42   0000:00:07.0    net_ixgbe
 Ethernet1/2         eth2       48:9b:d5:eb:1e:43   0000:00:08.0    net_ixgbe
```

    Compare this to the switch mapping by the ExtremeXOS CLI.

17. Inspect the ARP table:

```
# show iparp
VR          Destination      Mac                Age Static VLAN   VID   Port
VR-Default  192.168.1.100    48:9b:d5:eb:1e:42   6   NO   inside   10    57
VR-Default  10.10.100.100    48:9b:d5:eb:1e:43   1   NO   outside  100   58
```

18. Inspect the forwarding database:

```
# show fdb
MAC                 VLAN Name( Tag)    Age Flags   Port/Virtual Port List
-------------------------------------------------------------------------------
48:9b:d5:eb:1e:42    inside(0010) 0091   d     m          57
48:9b:d5:eb:1e:43    outside(0100) 0026  d     m          58
```

As each interface is mapped, activated, and used on the VM, they appear the IP ARP table and the switch forwarding database. Port management is shared with the front panel Management port and limited to 1 Gb/s. Ports 57 and 58 are connected directly to the forwarding plane on an X465. (A X695 has only one port connected to the forwarding plane and requires VLANs to instantiate multiple interfaces. Check the hardware guide for your switch for the number and port designations of the forwarding plane ports.)

ExtremeSwitching X695 series switches provide greater CPU and memory capacity, but have only one sideband port: port 63. This is a 10Gb/s port. The configuration of the firewall on the X695 requires the use of sub-interfaces or VLANs on the firewall to gain more than one firewall traffic port. The configuration is as above, but the two ports (Mgmt and port 63) are mapped. Mgmt provides the web UI and port 63 maps to Ethernet 1/1.

When using a ExtremeSwitching X695 switch for the above procedure, the following applies.

```
# sh vm PAN
VM Name:         PAN
    State:       Running
    Memory size: 8192 MB
    CPUs:        2
    Auto-start:  Disabled
    VNC:         127.0.0.1:1 (Port 5901)
    Disk:        vda
        Source:                  /mnt/vmdisk/.vm/PAN_PA-VM-KVM-9.1.2.qcow2
        Disk bus type:           virtio
        Allocated size in bytes: 64424509440 (60.00 GB)
        Physical size in bytes:  5405478912 (5.03 GB)
        Read requests:           48213
        Bytes read:              1188752896
        Write requests:          17699
        Bytes written:           1011475968
    Network interfaces:
        Attached switch ports:   mgmt,63
    CPU utilization:
        User:                    0.16%
        System:                  62.53%
    Memory utilization:
        Used:                    0.67 GB
        Available:               7.33 GB
```

The VLAN configuration of the sideband port is as follows:

```
# sh ports 63 vlan
         Untagged
Port     /Tagged   VLAN Name(s)
-------- --------  ------------------------------------------------------------
```

```
Insight  Untagged  Default
         Tagged    v1, v2
```

They are mapped directly to interface Ethernet 1/1 (Default VLAN on 63), sub-interface Ethernet 1/1.10 (VLAN 10, v1), sub-interface Ethernet 1/1.20 (VLAN 20, v2).

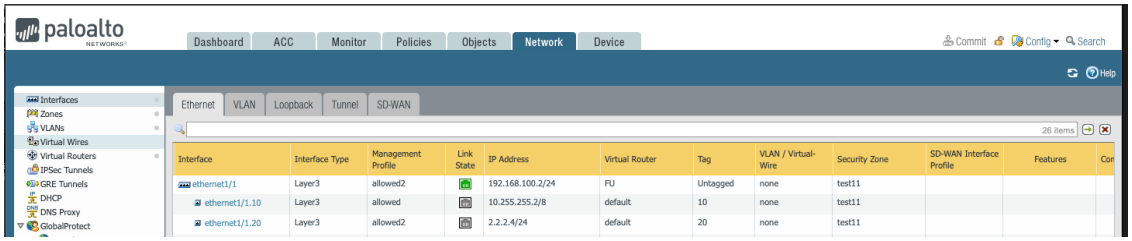
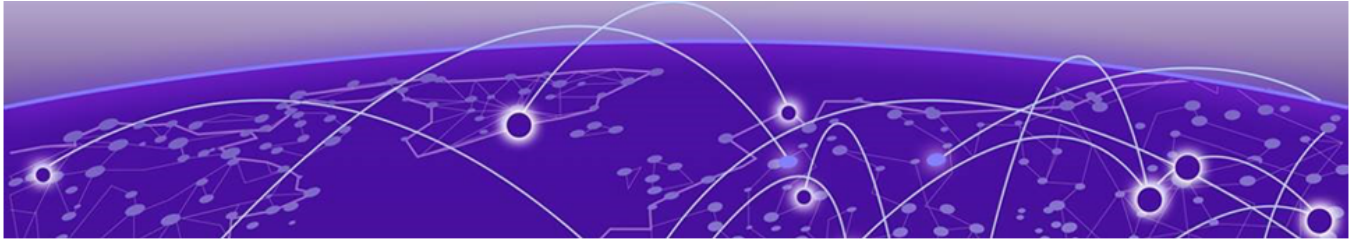


Figure 94: Palo Alto Firewall Interface Management Profile Window for X695

The total availability of bandwidth for this VM is one 10Gb/s port: 10Gb/s in, 10Gb/s out.



IPv4 Unicast Routing

IPv4 Unicast Overview on page 884
Configuring Unicast Routing on page 907
Displaying the Routing Configuration and Statistics on page 911
Routing Configuration Example on page 912
Address Resolution Protocol (ARP) and Neighbor Discovery (ND) on page 913
Proxy ARP on page 914
IPv4 Multinetting on page 916
DHCP/BOOTP Relay on page 922
DHCP Smart Relay on page 925
Broadcast UDP Packet Forwarding on page 927
IP Broadcast Handling on page 929
VLAN Aggregation on page 931
IP Network Address Translation (NAT) on page 934

This chapter assumes that you are already familiar with IP unicast routing. If not, refer to the following publications for additional information:

- RFC 1256—*ICMP (Internet Control Message Protocol)* Router Discovery Messages
- RFC 1812—Requirements for IP Version 4 Routers

Note

For more information on interior gateway protocols, see:

- Overview of RIP on page 1508
- OSPF
- IS-IS

For information on exterior gateway protocols, see BGP.

For more information on switch support for IPv6, IPv6 Unicast Routing on page 955.

IPv4 Unicast Overview

The switch provides full Layer 3, IPv4 unicast routing to all switches that run the Edge (or higher) or Base (or higher) license. (See the Switch Engine v33.6.1 Licensing Guide

document.) It exchanges routing information with other routers on the network using one of the following routing protocols:

- *RIP (Routing Information Protocol)*
- *OSPF (Open Shortest Path First)*
- *BGP (Border Gateway Protocol)*
- Intermediate System-Intermediate System (ISIS)

The switch dynamically builds and maintains a set of routing tables and determines the best path for each of its routes. Each host using the IP unicast routing functionality of the switch must have a unique IP address assigned. In addition, the default gateway assigned to the host must be the IP address of the router interface.

The ExtremeXOS software can provide both IPv4 and IPv6 routing at the same time. Separate routing tables are maintained for the two versions. Most commands that require you to specify an IP address can accept either an IPv4 or IPv6 address and act accordingly. Additionally, many of the IP configuration, enabling, and display commands have added tokens for IPv4 and IPv6 to clarify the version required. For simplicity, existing commands affect IPv4 by default and require you to specify IPv6, so configurations from an earlier release still correctly configure an IPv4 network.

Router Interfaces

The routing software and hardware routes IPv6 traffic between router interfaces. A router interface is either a virtual LAN (*VLAN (Virtual LAN)*) that has an IP address assigned to it, or a Layer 3 tunnel.

As you create VLANs and tunnels with IP addresses, you can also choose to route (forward traffic) between them. Both the VLAN switching and IP routing function occur within the switch.

IPv4 and IPv6 interfaces can coexist on the same VLAN, allowing both IPv4 and IPv6 networks to coexist on the same Layer 2 broadcast domain.

Note
Each IP address and mask assigned to a VLAN must represent a unique IP subnet. You cannot configure the same IP address and subnet on different VLANs within the same .

GRE Tunnel

ExtremeXOS 15.4 and later supports creating a Generic Routing Encapsulation (GRE)-based IPv4 tunnel, and routing IPv4 traffic over it. This feature is supported on all platforms that have GRE tunneling support (see Switch Engine v33.6.1 Licensing Guide).

ExtremeXOS 31.3 and later supports GRE tunnel configuration on user VRs instead of only on the Default VR. You can also configure the L3/IP interface of GRE tunnels on different VRs than the VR for the endpoints (source and destination IP addresses) using

the revised `create tunnel` *`tunnel_name`* **`gre destination`** *`destination-address`* **`source`** *`source-address`* {**`vr`** *`vr_name`*} {**`payload-vr`** *`payload_vr_name`*} command.

Note
Only one BGP session is recommended over a GRE tunnel that is configured in user VRs.

Note
The tunnel can be configured on user or default VR, but cannot be configured on VR-control and VR-mgmt.

As switch administrator, you can configure a GRE tunnel by supplying the local source IPv4 address and the destination IPv4 address. When configured, traffic can be redirected through the GRE tunnel. The TTL value of the outer IPv4 header is set to 255, and the DSCP value is copied from the inner IPv4 header, the same as for the IPv6 tunnels. The encapsulated packets do not include the GRE checksum option; however if received with a checksum they are verified by the software, and then dropped if incorrect. The GRE module is capable of dealing with *RFC 1701* neighbor options, with the exception of the router option. Packets with this option set are dropped. However, hardware does not support any options in the GRE header. If any of these options are set, the packet is either dropped, or sent to the CPU for processing. Since the key option of GRE tunnel is not configured, the GRE module only accepts GRE packets with a key value of 0, if present, and drops packets with other key values.

Note
GRE tunnels are IP tunnels, which require L3 function. L3 features are supported with an Edge or Base license or greater. All of the supported platforms' default license is Edge\Base or greater, which include L3 features.

All nodes in a stack need to support GRE tunnels, or else the feature cannot be configured/enabled. When all nodes in the system are capable of running GRE, new GRE tunnels can be created. If a new node is added to a stack that does not support GRE, when it is powered on, a log message is generated indicating that the node is not compatible with GRE, and should be removed. If a stack boots up with both a GRE configuration and a GRE-incapable node, all GRE tunnels are put in a system disabled state. This is done to prevent the node from continuously rebooting. The `show [{`**`tunnel`**`} {`*`tunnel_name`*`} | `**`tunnel`** `{`**`vr`** `[`*`vrname`* `|` **`all`**`]} {`**`payload-vr`** *`payload_vrname`*`} {`**`detail`**`]` command displays this.

IP Maximum Transmission Unit Configuration

GRE tunnel also supports IP Maximum Transmission Unit (MTU) configuration. When the IP packet size exceeds the configured IP MTU value on a tunnel, the Path MTU (PMTU) mechanism can be used to notify the sender of the traffic that the packet size needs to be reduced.

Use the `configure tunnel` *`tunnel_name`* **`ip mtu`** *`mtu`* command to set the MTU value. The default is 1476.

This feature is only available on ExtremeSwitching X465, and X695 series switches.

TCP Maximum Segment Size Adjustment

Starting with ExtremeXOS 31.6, GRE tunnels support TCP Maximum Segment Size (MSS) adjustment. Turning on the TCP MSS adjustment on a GRE tunnel redirects TCP SYN and SYN ACK packets going into the GRE tunnel to the Control Protocol (CP) via the ACL implemented on fastpath. When passing the IP stack on CP, the netfilter rule is applied to those packets. If the TCP MSS value in the option is greater than the configured adjustment, the TCP MSS option value is rewritten to the configured one. However, if the option value is less than the configured value, there is no change. Doing this helps avoid end-to-end over GRE tunnel application failure.

When the TCP MSS adjustment is turned off, there is no interference on TCP SYN and SYN ACK packets going to the GRE tunnel since the ACL is removed. Those packets are fastpath forwarded.

You can turn on or off TCP MSS adjustment using the `configure tunnel` *`tunnel_name`* **`ip tcp adjust-mss`** `[`**`off`** `|` **`on`** *`tcp_mss_value`*`]` command for IPv4, or the `configure tunnel` *`tunnel_name`* **`ipv6 tcp adjust-mss`** `[`**`off`** `|` **`on`** *`tcp_mss_value`*`]` command for IPv6. The *`tcp_mss_value`* `must be specified when turning the adjusment on.`

IPv6 Transport in IPv4 GRE Tunnel

Version 32.3 adds the ability to transport IPv6 payloads in IPv4 GRE tunnel, allowing the GRE tunnel to be configured with multiple IPv6 addresses. ECMP is also supported for IPv6 destinations reachable via GRE tunnel. IPv6 forwarding can be enabled on the GRE tunnel that has an existing IPv6 address configured, but only one ECMP Nexthop is installed in the hardware.

Use the `configure tunnel` *`tunnel_name`* **`ipv6 tcp adjust-mss`** `[`**`off`** `|` **`on`** *`tcp_mss_value`*`]` command to configure the IPv6 interface address(es) and IPv6 TCP MSS GRE tunnel.

Supported Platforms

All Universal platforms.

GRE Tunnel Example Configuration

This example shows how the GRE tunnel feature could be configured.

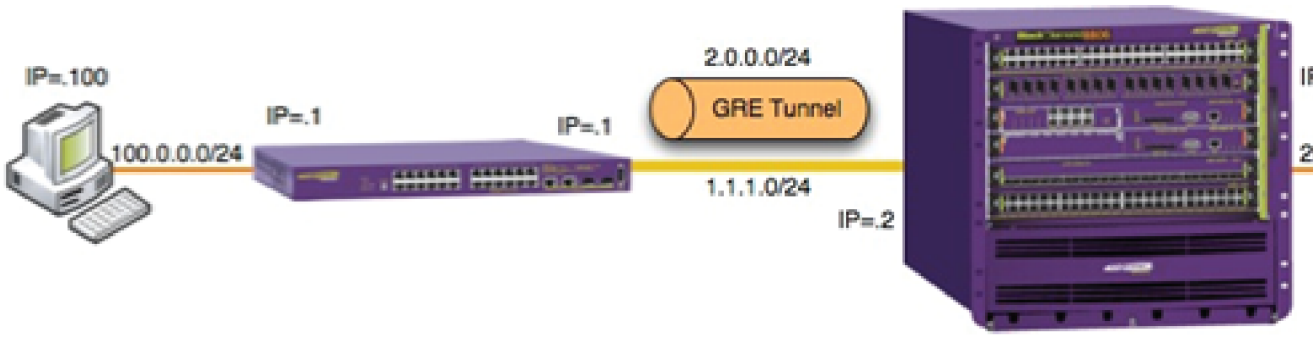


Figure 95: GRE Tunnel Configuration

Switch 1 Configuration

```
configure default del port all
create vlan inet
configure vlan inet add port 24
configure vlan inet ipa 1.1.1.1/24
create vlan users
configure vlan users add port 1
configure vlan users ipa 100.0.0.1/24
create tunnel mytunnel gre destination 1.1.1.2 source 1.1.1.1
configure tunnel "mytunnel" ipaddress 2.0.0.1/24
configure iproute add 200.0.0.0/24 2.0.0.2
enable ipforwarding
```

Switch 2 Configuration

```
unconfigure switch all
configure default del port all
create vlan inet
configure vlan inet add port 10:1
configure vlan inet ipa 1.1.1.2/24
create vlan users
configure vlan users add port 10:2
configure vlan users ipa 200.0.0.1/24
create tunnel mytunnel gre destination 1.1.1.1 source 1.1.1.2
configure tunnel "mytunnel" ipaddress 2.0.0.2/24
configure iproute add 100.0.0.0/24 2.0.0.1
enable ipforwarding
```

Populating the Routing Tables

The switch maintains a set of IP routing tables for both network routes and host routes. Some routes are determined dynamically from routing protocols, and some routes are manually entered. When multiple routes are available to a destination, configurable options such as route priorities, route sharing, and compressed routes are considered when creating and updating the routing tables.

Dynamic Routes

Dynamic routes are typically learned by enabling the *RIP*, *OSPF*, IS-IS or *BGP* protocols, and are also learned from *ICMP* redirects exchanged with other routers. These routes are called dynamic routes because they are not a permanent part of the configuration. The routes are learned when the router starts up and are dynamically updated as the network changes. Older dynamic routes are aged out of the tables when an update for the network is not received for a period of time, as determined by the routing protocol.

Once a routing protocol is configured, dynamic routes require no configuration and are automatically updated as the network changes.

Static Routes

Static routes are routes that are manually entered into the routing tables and are not advertised through the routing protocols. Static routes can be used to reach networks that are not advertised by routing protocols and do not have dynamic route entries in the routing tables. Static routes can also be used for security reasons, to create routes that are not advertised by the router.

Static routes are configured in the ExtremeXOS software, remain part of the configuration when the switch is rebooted, and are immediately available when the switch completes startup. Static routes are never aged out of the routing table, however, the Bidirectional Forwarding Detection (BFD) feature, can be used to bring down static routes when the host link fails.

Without BFD, static routes always remain operationally active because there is no dynamic routing protocol to report network changes. This can lead to a black hole situation, where data is lost for an indefinite duration. Because upper layer protocols are unaware that a static link is not working, they cannot switch to alternate routes and continue to use system resources until the appropriate timers expire.

With BFD, a static route is marked operationally inactive if the BFD session goes down. Upper layer protocols can detect that the static route is down and take the appropriate action.

A default route is a type of static route that identifies the default router interface to which all packets are routed when the routing table does not contain a route to the packet destination. A default route is also called a default gateway.

ExtremeXOS Resiliency Enhancement for IPv4 Static Routes

The ExtremeXOS Resiliency Enhancement feature provides a resilient way to use *ECMP (Equal Cost Multi Paths)* to load balance IPv4 traffic among multiple servers or other specialized devices. ExtremeXOS automatically manages the set of active devices using ECMP static routes configured with ping protection to monitor the health of these routes. Such servers or specialized devices do not require special software to support Bidirectional Forwarding Detection (BFD), or IP routing protocols such as *OSPF*, or proprietary protocols to provide keepalive messages. ExtremeXOS uses industry-standard and required protocols *ICMP*/ARP for IPv4 to accomplish the following automatically:

- Initially verify devices and activate their static routes, without waiting for inbound user traffic, and without requiring configuration of device MAC addresses.
- Detect silent device outages and inactivate corresponding static routes.
- Reactivate static routes after device recovery, or hardware replacement with a new MAC address.

ExtremeXOS currently supports similar protection and resiliency using Bidirectional Forwarding Detection (BFD) on IPv4 static routes. However, BFD can only be used when the local and remote device both support BFD.

Multiple Routes

When there are multiple, conflicting choices of a route to a particular destination, the router picks the route with the longest matching network mask. If these are still equal, the router picks the route using the following default criteria (in the order specified):

- Directly attached network interfaces
- Static routes
- *ICMP* redirects

- Dynamic routes
- Directly attached network interfaces that are not active.

Note
If you define multiple default routes, the route that has the lowest metric is used. If multiple default routes have the same lowest metric, the system picks one of the routes.

The criteria for choosing from multiple routes with the longest matching network mask is set by choosing the relative route priorities.

Relative Route Priorities

Table 50 lists the relative priorities assigned to routes depending on the learned source of the route.

Note
Although these priorities can be changed, do not attempt any manipulation unless you are expertly familiar with the possible consequences.

Table 50: Relative Route Priorities

Route Origin	Priority
Direct	10
BlackHole	50
Static	1100
HostMobility	1150
ICMP	1200
Auto-peering	1699
OSPFv3Intra	2200
OSPFv3Inter	2300
IS-IS	2350
IS-IS L1	2360
IS-IS L2	2370
RIPng	2400
OSPFv3 ASExt	3100
OSPFv3 Extern1	3200
OSPFv3 Extern2	3300
IS-IS L1Ext	3400
IS-IS L2Ext	3500
EBGP	1700
IBGP	1900

To change the relative route priority, use the following command:

```
configure iproute ipv6 priority [auto-peering | ripng | blackhole |
icmp | host-mobility | static | ospfv3-intra | ospfv3-inter | ospfv3-as-
external | ospfv3-extern1 | ospfv3-extern2 | isis |isis-level-1 | isis-
level-2 | isis-level-1-external | isis-level-2-external | ebgp | ibgp]
priority {vr vr_name}
```

Route priority configurable value is between 11-65,534.

IP Route Sharing and ECMP

IP route sharing allows a switch to communicate with a destination through multiple equal-cost routes. In *OSPF*, *BGP*, and IS-IS, this capability is referred to as *ECMP* routing.

Without IP route sharing, each IP route entry in the routing tables lists a destination subnet and the next-hop gateway that provides the best path to that subnet. Every time a packet is forwarded to a particular destination, it uses the same next-hop gateway.

With IP route sharing, an additional ECMP table lists up to 2, 4, 8, 16, 32, or 64 next-hop gateways (depending on the platform and feature configuration) for each route in the routing tables. When multiple next-hop gateways lead to the same destination, the switch can use any of those gateways for packet forwarding. IP route sharing provides route redundancy and can provide better throughput when routes are overloaded.

The gateways in the ECMP table can be defined with static routes (up to 64 way), or they can be learned through the OSPF, BGP, or IS-IS protocols (64-way for OSPFv2, *OSPFv3 (Open Shortest Path First version 3)*, and BGP; and 8-way for IS-IS). For more information on the ECMP table, see ECMP Hardware Table on page 905.

Note
BGP does not use ECMP by default, so if you require that functionality you must explicitly issue the command `configure bgp maximum-paths max-paths` with a value greater than 1.

Equal-cost Multi-path Routing (ECMP) Hashing Alternatives for IPv4/IPv6

You can dynamically choose between the existing *ECMP* hash method ("default") and the “custom” hash method. Within each of these two hash methods, several hash algorithm options are available to vary traffic distribution among multiple equal-cost IP gateways.

To configure the hash method, use the following command:

```
configure iproute sharing hash-method custom {hash-algorithm [xor |
crc-16 | crc-32 [lower | upper]]}
```

Compressed Routes

Compressed routes allow you to reduce the number of routes that are installed in the hardware routing tables. The switch uses hardware routing tables to improve packet forwarding performance. The switch can use both hardware and software to

forward packets, but packet forwarding without software processing is faster. The hardware routing tables have less storage space than the software, so compressed routes conserve resources and improve scaling

The compressed route feature allows you to install less specific routes in the table, when overlapping routes with same next-hop exist. This route pruning technique is implemented as part of the Route Manager (RtMgr) process.

When a route is added, deleted or updated, the pruning algorithm is applied to see if the new route and/or its immediate children can be compressed or uncompressed as follows:

- If the parent node (immediate less specific route) of the newly added IP prefix has the same gateway as the new IP prefix, the newly added prefix is compressed.
- If the gateways of the newly added IP prefix and its immediate children are the same, the child nodes are compressed.
- If the gateways of the newly added IP prefix and its immediate children are not the same, and the child nodes had been previously compressed, the child nodes are uncompressed.

Event Log Messages

MPLS (Multiprotocol Label Switching) has full EMS logging capabilities to capture error as well as debug information. Messages are logged with the MPLS component identifier.

Exceptional Scenarios

This section explains instances of exceptional route compression behavior.

- When a node does not have any best route.

  Consider the routing table shown in Table 51. When a node does not have any best route, children are uncompressed, if they were already compressed. Also this node is uncompressed, if it had previously been compressed.

Table 51: Route Manager's Table When There is No Best Route for a Node

Prefix	Gateway	Number of best paths	Compressed?
192.0.0.0/8	10.203.174.68	1	No
192.168.0.0/16	10.203.174.68	0	No
192.168.224.0/24	10.203.174.68	1	No
192.168.225.0/24	10.203.174.68	1	No

- When a node contains only a multicast route.

  Route compression is applied to unicast routes only. If a node contains only a multicast route, the compression algorithm is not applied to the node. Therefore

multicast nodes are considered as nodes with no best unicast routes as shown in Table 52.

Table 52: Route Manager's Table When a Node Contains Only a Multicast Route

Prefix	Gateway	Unicast/Multicast	Compressed?
192.0.0.0/8	10.203.174.68	Unicast Route	No
192.168.0.0/16	10.203.174.68	Multicast Route	No
192.168.224.0/24	10.203.174.68	Unicast Route	No
192.168.225.0/24	10.203.174.68	Unicast Route	No

ECMP Handling When IP Route Sharing Is Enabled

The nodes that have *ECMP* table entries are compressed only if the following conditions are met; otherwise, potential sub-optimal forwarding occurs:

- The number of ECMP gateways for a given node must match the number of ECMP gateways in its parent node.
- A given node's set of gateways must match its parent's set of gateways.

Table 53 shows how compression is applied for the nodes with ECMP table entries when IP route sharing is enabled. Sample routes with ECMP table entries are taken for illustration. The Reason field in the table provides information about why the compression is applied or not applied for the node.

Table 53: Route Manager's Table When IP Route Sharing is Enabled

Prefix	Gateways	Compressed ?	Reason
20.0.0.0/8	Gw1: 30.1.10.1, Gw2: 50.1.10.1	NO	This is the top node.
20.1.10.0/24	Gw1: 30.1.10.1	NO	Number of gateways did not match. This node has only one gateway, while the parent node has two.
20.2.10.0/24	Gw1: 30.1.10.1, Gw2: 60.1.10.1	NO	Number of gateways match. But one of the ECMP paths (gateway 60.1.10.1) does not match with its parent's ECMP paths.
20.3.10.0/24	Gw1: 30.1.10.1, Gw2: 50.1.10.1	YES	Number of gateways matches with its parent. Also all the gateways match with parent.
20.4.10.0/24	Gw1: 30.1.10.1, Gw2: 50.1.10.1 Gw3: 60.1.10.1	NO	Number of gateways does not match with its parent.
20.1.10.44/32	Gw1: 30.1.10.1 Gw2: 50.1.10.1	NO	Number of gateways did not match. [This node has ECMP table entries, but parent node 20.1.10.0 does not have an ECMP table entry.]

Table 54 shows only uncompressed routes.

Table 54: HAL(TCAM)/Kernel Routing Table When IP Route Sharing is Enabled

Prefix	Gateway
20.0.0.8/16	Gw1: 30.1.10.1, Gw2: 50.1.10.1
20.1.10.0/24	Gw1: 30.1.10.1
20.2.10.0/24	Gw1: 30.1.10.1, Gw2: 60.1.10.1
20.4.10.0/24	Gw1: 30.1.10.1, Gw2: 50.1.10.1 Gw3: 60.1.10.1
20.1.10.44/32	Gw1: 30.1.10.1 Gw2: 50.1.10.1

Sample output is shown below:

```
* (debug) BD-12804.9 # enable iproute sharing
* (debug) BD-12804.10 # show iproute
Ori Destination      Gateway       Mtr  Flags       VLAN   Duration
#s  Default Route    12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#s  Default Route    12.1.10.12    1    UG---S-um--f v1     0d:19h:14m:58s
#d  12.1.10.0/24     12.1.10.62    1    U------um--f v1     0d:20h:1m:3s
d   16.1.10.0/24     16.1.10.62    1    -------um--- v16    0d:20h:1m:4s
#d  22.1.10.0/24     22.1.10.62    1    U------um--f v2     0d:20h:1m:4s
#s  33.33.33.0/24    12.1.10.25    1    UG---S-um--f v1     0d:20h:1m:3s
#s  55.0.0.0/8       12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#s  55.0.0.0/8       22.1.10.33    1    UG---S-um--f v2     0d:20h:1m:3s
#s  55.2.1.1/32      12.1.10.22    1    UG---S-um--f v1     0d:20h:1m:3s
#s  55.5.5.1/32      12.1.10.44    1    UG---S-um--f v1     0d:20h:1m:3s
#s  66.0.0.0/8       12.1.10.12    1    UG---S-um--f v1     0d:20h:1m:3s
#s  66.0.0.0/16      12.1.10.12    1    UG---S-um--c v1     0d:20h:1m:3s
#d  70.1.10.0/24     70.1.10.62    1    U------um--f v7     0d:20h:1m:4s
#s  78.0.0.0/8       12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#s  79.0.0.0/8       12.1.10.10    1    UG---S-um--c v1     0d:20h:1m:3s
#s  79.0.0.0/8       12.1.10.12    1    UG---S-um--c v1     0d:20h:1m:3s
#s  80.0.0.0/8       12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#d  80.1.10.0/24     80.1.10.62    1    U------um--f v8     0d:20h:1m:4s
#s  81.0.0.0/8       12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#s  81.0.0.0/8       12.1.10.12    1    UG---S-um--f v1     0d:20h:1m:3s
#s  81.0.0.0/8       12.1.10.13    1    UG---S-um--f v1     0d:20h:1m:3s
#s  82.0.0.0/8       12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#s  83.0.0.0/8       12.1.10.10    1    UG---S-um--f v1     0d:20h:1m:3s
#d  91.1.10.0/24     91.1.10.62    1    U------um--f v9     0d:20h:1m:4s
#d  92.1.10.0/24     92.1.10.62    1    U------um--f v10    0d:20h:1m:6s
#d  93.1.10.0/24     93.1.10.62    1    U------um--f v11    0d:20h:1m:6s

Origin(Ori): (b) BlackHole, (be) EBGP, (bg) BGP, (bi) IBGP, (bo) BOOTP
(ct) CBT, (d) Direct, (df) DownIF, (dv) DVMRP, (e1) ISISL1Ext
(e2) ISISL2Ext, (h) Hardcoded, (i) ICMP, (i1) ISISL1 (i2) ISISL2
(is) ISIS, (mb) MBGP, (mbe) MBGPExt, (mbi) MBGPInter, (mp) MPLS Lsp
(mo) MOSPF (o) OSPF, (o1) OSPFExt1, (o2) OSPFExt2
(oa) OSPFIntra, (oe) OSPFAsExt, (or) OSPFInter, (pd) PIM-DM, (ps) PIM-SM
(r) RIP, (ra) RtAdvrt, (s) Static, (sv) SLB_VIP, (un) UnKnown
(*) Preferred unicast route (@) Preferred multicast route
(#) Preferred unicast and multicast route

Flags: (B) BlackHole, (D) Dynamic, (G) Gateway, (H) Host Route
(L) Matching LDP LSP, (l) Calculated LDP LSP, (m) Multicast
(P) LPM-routing, (R) Modified, (S) Static, (s) Static LSP
(T) Matching RSVP-TE LSP, (t) Calculated RSVP-TE LSP, (u) Unicast, (U) Up
(f) Provided to FIB (c) Compressed Route
```

```
Mask distribution:
2 default routes               12 routes at length 8
1 routes at length 16           9 routes at length 24
2 routes at length 32

Route Origin distribution:
8 routes from Direct              18 routes from Static

Total number of routes = 26
Total number of compressed routes = 3
```

ECMP Handling When IP Route Sharing Is Disabled

If IP route sharing is disabled, the first best route is installed in the hardware table, if multiple best routes are available. Hence the compression algorithm considers the first best route for *ECMP* cases. As shown in the following table, when IP route sharing is disabled, all routes are compressed, except the first one in this case.

Table 55: Route Manager's Table When IP Route Sharing is Disabled

Prefix	Gateways	Compressed?
20.0.0.0/8	Gw1: 30.1.10.1, Gw2: 50.1.10.1	NO
20.1.10.0/24	Gw1: 30.1.10.1	YES
20.2.10.0/24	Gw1: 30.1.10.1, Gw2: 60.1.10.1	YES
20.3.10.0/24	Gw1: 30.1.10.1, Gw2: 50.1.10.1	YES
20.4.10.0/24	Gw1: 30.1.10.1, Gw2: 50.1.10.1 Gw3: 60.1.10.1	YES
20.1.10.44/32	Gw1: 30.1.10.1 Gw2: 50.1.10.1	YES

Table 56: HAL(TCAM)/Kernel Routing Table When IP Route Sharing is Disabled

Prefix	Gateway
20.0.0.8/16	Gw1: 30.1.10.1,

Sample output is shown below:

```
* (debug) # disable iproute sharing
* (debug) # show iproute
Ori Destination        Gateway         Mtr  Flags        VLAN       Duration
#s  Default Route      12.1.10.10      1    UG---S-um--f v1         0d:19h:58m:58s
#s  Default Route      12.1.10.12      1    UG---S-um--- v1         0d:19h:12m:53s
#d  12.1.10.0/24       12.1.10.62      1    U------um--f v1         0d:19h:58m:59s
d  16.1.10.0/24       16.1.10.62      1    -------um--- v16        0d:19h:58m:59s
#d  22.1.10.0/24       22.1.10.62      1    U------um--f v2         0d:19h:58m:59s
#s  33.33.33.0/24      12.1.10.25      1    UG---S-um--f v1         0d:19h:58m:58s
#s  55.0.0.0/8         12.1.10.10      1    UG---S-um--c v1         0d:19h:58m:58s
#s  55.0.0.0/8         22.1.10.33      1    UG---S-um--- v2         0d:19h:58m:58s
#s  55.2.1.1/32        12.1.10.22      1    UG---S-um--f v1         0d:19h:58m:58s
#s  55.5.5.1/32        12.1.10.44      1    UG---S-um--f v1         0d:19h:58m:58s
#s  66.0.0.0/8         12.1.10.12      1    UG---S-um--f v1         0d:19h:58m:58s
#s  66.0.0.0/16        12.1.10.12      1    UG---S-um--c v1         0d:19h:58m:58s
```

```
#d  70.1.10.0/24       70.1.10.62      1    U------um--f v7          0d:19h:58m:59s
#s  78.0.0.0/8         12.1.10.10      1    UG---S-um--c v1          0d:19h:58m:58s
#s  79.0.0.0/8         12.1.10.10      1    UG---S-um--c v1          0d:19h:58m:58s
#s  79.0.0.0/8         12.1.10.12      1    UG---S-um--- v1          0d:19h:58m:58s
#s  80.0.0.0/8         12.1.10.10      1    UG---S-um--c v1          0d:19h:58m:58s
#d  80.1.10.0/24       80.1.10.62      1    U------um--f v8          0d:19h:58m:59s
#s  81.0.0.0/8         12.1.10.10      1    UG---S-um--c v1          0d:19h:58m:58s
#s  81.0.0.0/8         12.1.10.12      1    UG---S-um--- v1          0d:19h:58m:58s
#s  81.0.0.0/8         12.1.10.13      1    UG---S-um--- v1          0d:19h:58m:58s
#s  82.0.0.0/8         12.1.10.10      1    UG---S-um--c v1          0d:19h:58m:58s
#s  83.0.0.0/8         12.1.10.10      1    UG---S-um--c v1          0d:19h:58m:58s
#d  91.1.10.0/24       91.1.10.62      1    U------um--f v9          0d:19h:58m:59s
#d  92.1.10.0/24       92.1.10.62      1    U------um--f v10         0d:19h:59m:2s
#d  93.1.10.0/24       93.1.10.62      1    U------um--f v11         0d:19h:59m:2s

Origin(Ori): (b) BlackHole, (be) EBGP, (bg) BGP, (bi) IBGP, (bo) BOOTP
(ct) CBT, (d) Direct, (df) DownIF, (dv) DVMRP, (e1) ISISL1Ext
(e2) ISISL2Ext, (h) Hardcoded, (i) ICMP, (i1) ISISL1 (i2) ISISL2
(is) ISIS, (mb) MBGP, (mbe) MBGPExt, (mbi) MBGPInter, (mp) MPLS Lsp
(mo) MOSPF (o) OSPF, (o1) OSPFExt1, (o2) OSPFExt2
(oa) OSPFIntra, (oe) OSPFAsExt, (or) OSPFInter, (pd) PIM-DM, (ps) PIM-SM
(r) RIP, (ra) RtAdvrt, (s) Static, (sv) SLB_VIP, (un) UnKnown
(*) Preferred unicast route (@) Preferred multicast route
(#) Preferred unicast and multicast route

Flags: (B) BlackHole, (D) Dynamic, (G) Gateway, (H) Host Route
(L) Matching LDP LSP, (l) Calculated LDP LSP, (m) Multicast
(P) LPM-routing, (R) Modified, (S) Static, (s) Static LSP
(T) Matching RSVP-TE LSP, (t) Calculated RSVP-TE LSP, (u) Unicast, (U) Up
(f) Provided to FIB (c) Compressed Route

Mask distribution:
2 default routes               12 routes at length 8
1 routes at length 16           9 routes at length 24
2 routes at length 32
Route Origin distribution:
8 routes from Direct              18 routes from Static

Total number of routes = 26
Total number of compressed routes = 8
```

LSP Route Handling

An LSP route is compressed only in the following circumstances:

- If the parent node of the LSP route is also an LSP route
- If the LSP nexthop of the parent node matches with this node

Hardware Routing Table Management

The switch hardware can route traffic based on information stored in hardware and software routing tables. Routing tasks are completed much faster when the routes are already stored in hardware routing tables. When packets are routed using the hardware routing tables, this is called fast-path routing, because the packets take the fast path through the switch.

The switch hardware provides the fast path, and the ExtremeXOS software provides the routing management capabilities, including management of the hardware routing tables. The software collects all routing information, manages the routes according to the switch configuration, and stores the appropriate routes in the hardware routing

tables. When an IP unicast packet arrives and the destination IP address is not found in the hardware route tables, it is routed by software. The software processing takes more time than the fast path, so routing based on the software tables is called slow-path routing.

ExtremeXOS switches allow you to customize the software management of the hardware routing tables using the following hardware components:

- Extended IPv4 Host Cache on page 897
- ECMP Hardware Table on page 905

Extended IPv4 Host Cache

The extended IPv4 host cache feature provides additional, configurable storage space on L3-capable switches to store additional IPv4 hosts in the hardware routing tables. This feature is supported on all switches.

All switches support slow-path routing (using software routing tables), so adding more entries in the hardware routing table is a performance feature, which allows more hosts to benefit from fast-path routing. To use the extended IPv4 host cache feature effectively, it helps to understand how the hardware tables operate on the switches that support this feature. The hardware forwarding tables controlled by this feature store entries for the following:

- IPv4 local and remote hosts
- IPv4 routes
- IPv6 local hosts
- IPv6 routes
- IPv4 and IPv6 multicast groups

The extended IPv4 host cache feature works by customizing the forwarding table space allotted to these components.

Introduction to Hardware Forwarding Tables

The extended IPv4 host cache feature relates to the four hardware forwarding tables shown in Figure 96.

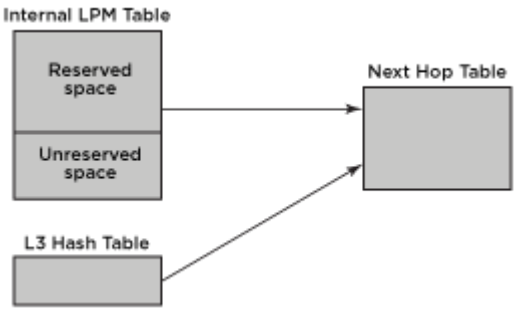


Figure 96: Hardware Forwarding Tables

The Longest Prefix Match (LPM) and Layer 3 (L3) Hash tables store host and route information for fast-path forwarding. When the switch locates a route or host in one of these tables, it follows a table index to the Next Hop table, which contains MAC address

and egress port information that is shared by the hosts and routes in the other tables. The hardware routing table capacity is partly determined by the capacity of the Next Hop table. The Next Hop table capacity is smaller than the combined capacity of the other tables because typically, multiple routes and hosts share each Next Hop table entry. When the other tables map to many different next hop entries, the Next Hop table can limit the total number of hosts and routes stored in hardware.

On most platforms, the L3 Hash table is smaller than the LPM tables. Because the L3 Hash table is the only table that can store IPv4 and IPv6 multicast entries and IPv6 local hosts, and because of the way the L3 Hash table is populated, forwarding table capacity and forwarding performance can be improved by allocating space for storing IPv4 local and remote host entries in the LPM tables.

The extended IPv4 host cache feature specifically allows you to define the number of entries that are reserved in the LPM tables for IPv4 and IPv6 routes. The unreserved entries are available for IPv4 local and remote hosts. IPv4 hosts can also occupy unused areas of the L3 Hash table, and when necessary, unused space in the reserved section of the LPM tables. The maximum number of hosts that can be stored in the hardware routing tables depends on the configuration and usage of the tables, but the number of local IPv4 hosts and gateways is ultimately limited to the size of the Next Hop table minus three reserved entries.

LPM Table Management

The internal LPM tables are provided on all platforms.

The ExtremeXOS software manages the content of the hardware tables based on the configuration specified by the following commands:

`configure iproute reserved-entries [` *num_routes_needed* `|` **`maximum`** `|` **`default`** `] slot [`**`all`** `|` *slot_num*`]`

The `configure iproute reserved-entries` command configures the LPM tables.

The `configure forwarding internal-tables [` **`l2-and-l3`** `|` **`more`** `[`**`l2`** `|` **`l3-and-ipmc`** `|` **`routes`** `{`**`ipv6-mask-length`** `[`**`64`** `|` **`128`**`]}]]` command provides the ability to support additional L2 and L3 hosts, routes, or multicast table entries.

Starting with ExtremeXOS 22.5, the ipv6-mask-length option provides greater hardware route scale and IP route sharing (*ECMP*) support for IPv6 “long-mask routes”, meaning IPv6 subnets with mask lengths 65–128 bits. This provides additional scale and resilience for IPv6 host routes whose mask length is 128 bits. Increasing scale and providing ECMP for IPv6 mask 65–128 routes decreases IPv4 route scale. The default IPv6 mask length is 64.

IPv6 hardware and slowpath forwarding are supported on user-created Virtual Routers, and IPv6 tunnels are only supported on *VR-Default*.

The size of the internal LPM tables, and the size of the L3 Hash and Next Hop tables are fixed for some platforms. The following tables list the hardware capacity for each of the tables shown in Figure 96 on page 897.

The ExtremeSwitching 5520 series switches have hardware forwarding tables that can be partitioned in a flexible manner. The ExtremeSwitching 5520 series switches have the following configurable internal tables.

Table 57: ExtremeSwitching 5520 Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
Raw L3 Hash table	64K	112K	16K	16K	16K
Max IPv4 Unicast in L3 Hash	40K	64K	16K	16K	16K
Max IPv6 Unicast in L3 Hash	16K	16K	8K	8K	8K
Next Hops for Underlay/ Normal	16K	16K	16K	16K	16K
IPv4 routes (LPM entries in HW)	16K	16K	16K	80K	40K
IPv6 routes (LPM entries in HW)	8K	8K	8K	40K	8K
IPv4 ARP entries in hardware with minimum LPM routes (assumes 75% utilization of L3 hash table)	40K	58K	16K	12K	12K
IPv4 ARP entries in hardware with minimum LPM routes (assumes max 100% utilization of L3 hash table)	48K	72K	16K	16K	16K
IPv4 ARP entries in hardware with maximum LPM routes (assumes 75% utilization of L3 hash table)	30K	48K	12K	12K	12K
IPv4 ARP entries in hardware with maximum LPM routes (assumes 100% utilization of L3 hash table)	40K	64K	16K	16K	16K
IPv4 remote hosts in hardware with zero LPM routes (assumes 75% utilization of L3 hash table)	64K	100K	28K	12K	12K
IPv6 host entries in hardware (assumes 75% utilization of L3 hash table)	16K	16K	6K	6K	6K
IP router interfaces	4K	4K	4K	4K	4K

Table 57: ExtremeSwitching 5520 Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables (continued)

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
IP multicast groups	8K	8K	8K	8K	8K
IP-multicast (s,v,g) entries (will depend on hash utilization)	32K	56K	8K	8K	8K

The ExtremeSwitching 5720 series switches have hardware forwarding tables that can be partitioned in a flexible manner. The ExtremeSwitching 5720 5G models have the following configurable internal tables.

Table 58: ExtremeSwitching 5720 5G Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
Raw L3 Hash table	96K	160K	32K	32K	32K
Max IPv4 Unicast in L3 Hash	60K	92K	32K	32K	32K
Max IPv6 Unicast in L3 Hash	24K	24K	16K	16K	16K
Next Hops for Underlay/ Normal	24K	24K	24K	24K	24K
IPv4 routes (LPM entries in HW)	16K	16K	16K	163K	107K
IPv6 routes (LPM entries in HW)	8K	8K	8K	107K	16K
IPv4 ARP entries in hardware with minimum LPM routes (assumes 75% utilization of L3 hash table)	55K	79K	24K	24K	24K
IPv4 ARP entries in hardware with minimum LPM routes (assumes max 100% utilization of L3 hash table)	68K	100K	24K	24K	24K
IPv4 ARP entries in hardware with maximum LPM routes (assumes 75% utilization of L3 hash table)	45K	69K	24K	24K	24K

Table 58: ExtremeSwitching 5720 5G Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables (continued)

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
IPv4 ARP entries in hardware with maximum LPM routes (assumes 100% utilization of L3 hash table)	60K	92K	24K	24K	24K
IPv4 remote hosts in hardware with zero LPM routes (assumes 75% utilization of L3 hash table)	88K	136K	40K	24K	24K
IPv6 host entries in hardware (assumes 75% utilization of L3 hash table)	24K	24K	12K	12K	12K
IP router interfaces	12286	12286	12286	12286	12286
IP multicast groups	8K	8K	8K	8K	8K
IP-multicast (s,v,g) entries (will depend on hash utilization)	48K	80K	16K	16K	16K

The ExtremeSwitching 5720 10G models have the following configurable internal tables.

Table 59: ExtremeSwitching 5720 10G Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
Raw L3 Hash table	160K	288K	32K	32K	32K
Max IPv4 Unicast in L3 Hash	140K	204K	32K	32K	32K
Max IPv6 Unicast in L3 Hash	80K	120K	16K	16K	16K
Next Hops for Underlay/ Normal	120K	16K	16K	16K	16K
IPv4 routes (LPM entries in HW)	24K	24K	24K	288K	213K
IPv6 routes (LPM entries in HW)	12K	12K	12K	213K	24K

Table 59: ExtremeSwitching 5720 10G Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables (continued)

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
IPv4 ARP entries in hardware with minimum LPM routes (assumes 75% utilization of L3 hash table)	120K	168K	48K	24K	24K
IPv4 ARP entries in hardware with minimum LPM routes (assumes max 100% utilization of L3 hash table)	152K	216K	56K	32K	32K
IPv4 ARP entries in hardware with maximum LPM routes (assumes 75% utilization of L3 hash table)	105K	153K	24K	24K	24K
IPv4 ARP entries in hardware with maximum LPM routes (assumes 100% utilization of L3 hash table)	140K	204K	32K	32K	32K
IPv4 remote hosts in hardware with zero LPM routes (assumes 75% utilization of L3 hash table)	144K	240K	48K	24K	24K
IPv6 host entries in hardware (assumes 75% utilization of L3 hash table)	53K	77K	12K	12K	12K
IP router interfaces	12286	12286	12286	12286	12286
IP multicast groups	32K	32K	32K	32K	32K
IP-multicast (s,v,g) entries (will depend on hash utilization)	80K	144K	16K	16K	16K

The Extreme 7520 and 7720 series switches have hardware forwarding tables that can be partitioned in a flexible manner. These switches have the following configurable internal tables.

Table 60: Extreme 7520, 7720 Hardware Routing Table Configuration Capacities for Platforms with Configurable L3 Internal Tables

L3 Characteristic	Internal Tables Configuration				
	l2-and-l3	more l3-and-ipmc	more l2	more routes (ALPM) w/ mask len. 64	more routes (ALPM) w/ mask len. 128
Max IPv4 Unicast in L3 Hash	84K	132K	16K	16K	16K
Max IPv6 Unicast in L3 Hash	56K	56K	8K	8K	8K
Next Hops for Underlay/ Normal	56K	56K	56K	56K	56K
IPv4 routes (LPM entries in HW)	16K	16K	16K	384K	256K
IPv6 routes (LPM entries in HW)	8K	8K	8K	256K	32K
IPv4 ARP entries in hardware with minimum LPM routes (assumes 75% utilization of L3 hash table)	78K	119K	44K	12K	12K
IPv4 ARP entries in hardware with maximum LPM routes (assumes 75% utilization of L3 hash table)	63K	99K	12K	12K	12K
IPv4 remote hosts in hardware with zero LPM routes (assumes 75% utilization of L3 hash table)	116K	88K	44K	12K	12K
IPv6 host entries in hardware (assumes 75% utilization of L3 hash table)	32K	56K	6K	6K	6K
IP router interfaces	4094	4094	4094	4094	4094
IP multicast groups	8K	8K	8K	8K	8K
IP-multicast (s,v,g) entries (will depend on hash utilization)	56K	104K	8K	8K	8K

Extended IPv4 Host Cache Management Guidelines

When configuring the extended IPv4 host cache feature, consider the following guidelines:

- The default option configures the switch to store entries for local and remote IPv4 hosts in the LPM tables.

- The maximum option reserves all space in the LPM tables for IPv4 and IPv6 routes. This option provides the maximum storage for IPv4 and IPv6 routes when you do not expect to store many IPv4 and IPv6 multicast and IPv6 hosts in the L3 Hash table.

Note
If no IPv4 route is found in the LPM table and IPv4 unicast packets are slow-path forwarded for a given remote host, an IPv4 entry is created for the remote host in either the L3 hash table or LPM table. The hardware does not cache entries for remote IPv6 hosts, so IPv6 routes take precedence over IPv4 routes.

IPv4 Host Entry Population Sequence

The ExtremeXOS software populates the hardware tables with IPv4 host entries by searching for available space in the following sequence:

1. Unreserved space in the LPM tables.
2. Available space in an L3 Hash table bucket.
3. Available space in the reserved section of the LPM table.
4. Space used by the oldest host entries in the LPM and L3 Hash tables.

The L3 Hash table is named for the hash function, which stores host and multicast entries based on an algorithm applied to the host IP address or multicast tuple (Source IP, Group IP, *VLAN* ID). The hash table stores entries in groups of 8 or 16 (depending on the hardware), and these groups are called buckets. When a bucket is full, any additional host or multicast addresses that map or hash to that bucket cannot be added. Another benefit of the extended IPv4 host cache feature is that you can reduce these conflicts (or “hash table collisions”), by making room for IPv4 hosts in the LPM table and reducing demand for the L3 Hash table.

A hardware-based aging mechanism is used to remove any remote IPv4 host entries that have not had IPv4 unicast packets forwarded to them in the previous hour. (Note that remote IPv4 hosts only need to be cached when all IPv4 routes do not fit within the number of routes reserved.) Aging helps to preserve resources for the hosts that are needed most. In a SummitStack, aging is performed independently for each stack node based on the ingress traffic for that node. Even with aging, it is still possible that the Next Hop table, LPM table, or L3 Hash bucket do not have space to accept a new host. In those cases, a least-recently used algorithm is used to remove the oldest host to make space for the new host in hardware.

Local IPv4 host entries are only subject to hardware-based aging if there has been a large amount of least-recently used replacement, indicating severe contention for HW table resources. Otherwise, local IPv4 host entries are retained just as in ExtremeXOS

releases prior to 12.1, based on whether IP ARP refresh is enabled or disabled, and the value for the configure iparp timeout command.

Note
Gateway entries are entries that represent routers or tunnel endpoints used to reach remote hosts. Gateway entries are not aged and are not replaced by IPv6 hosts or multicast entries in the L3 Hash table or by any entries requiring space in the Next Hop table. The software can move gateway entries from the LPM table to the L3 Hash table to make room for new reserved routes.

Calculating the Number of Routes Needed

Guidelines for calculating the number of routes to reserve are provided in the ExtremeXOS Command Reference description for the following command:

```
configure iproute reserved-entries [ num_routes_needed | maximum |
default ] slot [all | slot_num]
```

ECMP Hardware Table

IP route sharing and the *ECMP* hardware table are introduced in IP Route Sharing and ECMP on page 891. The following sections provide guidelines for managing the ECMP hardware table:

- ECMP Table Configuration Guidelines
- Troubleshooting: ECMP Table-Full Messages on page 906

ECMP Table Configuration Guidelines

The ECMP table contains gateway sets, and each gateway set defines the equal-cost gateways that lead to a destination subnet. When IP route sharing is enabled, subnet entries in the LPM table can be mapped to gateway set entries in the ECMP table, instead of to a single gateway within the LPM table.

For improved ECMP scaling, each LPM table entry points to a gateway set entry in the ECMP table. Each gateway set entry is unique and appears only once in the ECMP table.

Each gateway set entry for the platforms listed above is unique and appears only once in the ECMP table. Multiple LPM table entries can point to the same gateway set entry. This efficient use of the ECMP table creates more room in the ECMP table for additional gateway set entries. It also makes IP route sharing available to every entry in the LPM table.

The following command allows you to configure the maximum number of next-hop gateways for gateway sets in the ECMP table:

```
configure iproute sharing max-gateways max_gateways
```

Each gateway entry in a gateway set consumes ECMP table space. As the *max_gateways* value decreases, the ECMP table supports more gateway sets. If you configure the *max_gateways* value to 32, the switch supports route sharing through up to 32 gateways per subnet, but supports the smallest number of gateway sets. If you do

not need to support up to 32 different gateways for any subnet, you can decrease the *max_gateways* value to support more gateway sets.

To determine which gateways might be added to the ECMP table, consider how many local gateways are connected to the switch and can be used for ECMP, and consider the *max_gateways* value.

For example, suppose that you have four ECMP gateway candidates connected to the switch (labeled A, B, C, and D for this example) and the *max_gateways* option is set to 4. For platforms that allow a gateway set entry to support multiple subnets, this configuration could result in up to 11 gateway sets in the ECMP table: ABCD, ABC, ABD, ACD, BCD, AB, AC, AD, BC, BD, and CD.

If there are 4 gateways and you set *max-gateways* to 4, you can use the choose mathematical function to calculate the total number of gateway set possibilities as follows:

```
(4 choose 4) + (4 choose 3) + (4 choose 2) = 11
```

To calculate the number of gateway set possibilities for a given number of total gateways and a specific *max-gateways* value, use the choose function in the following formula:

```
(TGW choose MGW) + (TGW choose MGW-1) + ... + (TGW choose 2) = TGWsets
```

In the formula above, TGW represents the total local gateways, MGW represents the *max_gateways* value, and TGWsets represents the total gateway sets needed to support all possible shared paths.

To see if your platform supports the total gateway sets needed, do the following:

- Calculate the total ECMP gateway sets possible as described above.
- Compare your result to the IP route sharing (total combinations of gateway sets) capacities listed in the *ExtremeXOS Release Notes* to verify that the switch can support the desired number of gateway sets.

Troubleshooting: ECMP Table-Full Messages

If the ECMP table is full, no new gateway sets can be added, and IP forwarding is still done in hardware through one of the following:

- For platforms that allow a gateway set entry to support multiple subnets, forwarding can be done using an existing gateway set that is a partial subset of the unavailable gateway set. If the unavailable gateway set consists of N gateways, the subset used could include a range of gateways from N-1 gateways down to a single gateway. For example, if the ECMP table does not have room for a new gateway set using gateways E, F, G, and H, a partial subset such as EFG, EF, or E will be used.
- For platforms that require one gateway set entry per subnet, forwarding is done through a single gateway.

On ExtremeSwitching switches, an ECMP table-full condition produces the following message:

```
Noti:Card.IPv4FIB.ECMPTblFull Slot-1: IPv4 route can not use sharing on all its gateways.
Hardware ECMP Table full.  Packets are HW forwarded across a subset of gateways.   (Logged
at most once per hour.)
```

If the ECMP table-full message appears, consider the following remedies:

- Reduce the number of gateways adjacent to the switch used for IP route sharing.
- Monitor the switch to see if the condition is transient. For example, if the number of ECMP table entries temporarily increases due to a network event and then returns to within the supported range, a permanent change might not be required.
- Determine if IP route sharing to all gateways is required. Since traffic is still being forwarded in hardware using one or more gateways, the condition may be acceptable.

Configuring Unicast Routing

Configuring Basic Unicast Routing

To configure IP unicast routing on the switch:

1. Create and configure two or more *VLAN*s.
2. For each VLAN that participates in IP routing, assign an IP address, use the following command:

   `configure {vlan} vlan_name ipaddress [ipaddress {ipNetmask} | ipv6-link-local | {eui64} ipv6_address_mask]`

   `Ensure that each VLAN has a unique IP address.`
3. Configure a default route using the following command:

   `configure iproute add default gateway {metric} {multicast | multicast-only | unicast | unicast-only} {vr vrname}`

   `Default routes are used when the router has no other dynamic or static route to the requested destination.`
4. Turn on IP routing for one or all VLANs using the following command:

   `enable ipforwarding {ipv4 | broadcast | ignore-broadcast | fast-direct-broadcast} {vlan vlan_name}`
5. Configure the routing protocol, if required. For a simple network using RIP, the default configuration may be acceptable.
6. Turn on *RIP* or *OSPF* using one of the following commands:

   `enable rip`

   `enable ospf`

Adding a Default Route or Gateway

A default route or gateway defines a default interface to which traffic is directed when no specific routes are available. To add a default route, use the command:

```
configure iproute add default gateway {metric} {multicast | multicast-
only | unicast | unicast-only} {vr vrname}
```

Note
If you define a default route and subsequently delete the *VLAN* on the subnet associated with the default route, the invalid default route entry remains. You must manually delete the configured default route.

Configuring Static Routes

- To configure a static route, use the command:
  ```
  configure iproute add [ipNetmask | ip_addr mask] gateway {bfd}
  {metric} {multicast | multicast-only | unicast | unicast-only} {vlan
  egress_vlan} {vr vrname}
  ```

  Note
  Tracert might not always work if inter-VR routing is configured in one of the hops to the destination.

  Note
  When inter-vr routing is configured the gateway address should be different from *VLAN* IP of the switch and it should be reachable (ARP resolved) from the switch.

  The gateway address cannot be loop back address or any local address. A static route's next-hop (gateway) must be associated with a valid IP subnet and cannot use the same IP address as a local VLAN. An IP subnet is associated with a single VLAN by its IP address and subnet mask. If the VLAN is subsequently deleted, the static route entries using that subnet must be deleted manually.

  For Inter-VR routing, the egress VLAN name specified in the static route command may be a VLAN belonging to a VR different from the VR of the static route itself. When the VRs differ, Inter-VR routing of hardware and software forwarded packets is performed.

  This command can enable or disable BFD protection for one static route. However, this protection is not provided until the BFD client is enabled at each end of the route with the following command:

  ```
  enable iproute bfd {gateway} ip_addr {vr vrname}
  ```

- To disable BFD protection for a static route, use the following command:
  ```
  disable iproute bfd {gateway} ip_addr {vr vrname}
  ```

Configuring the Relative Route Priority

To change the relative route priority, use the following command:

```
configure iproute {ipv4} priority [auto-peering | blackhole | bootp
| ebgp |host-mobility | ibgp | icmp | isis | isis-level-1 | isis-
level-1-external | isis-level-2 | isis-level-2-external | mpls | ospf-
```

as-external | **ospf-extern1** | **ospf-extern2** | **ospf-inter** | **ospf-intra** | **rip** | **static** | **evpn** | **ospf-inter-vr** | **direct-inter-vr**] *priority* {**vr** *vrname*}

Configuring Hardware Routing Table Usage

Allowing individual local and remote IPv4 unicast hosts to occupy the unused portion of the Longest Prefix Match (LPM) table helps reduce Layer 3 hardware hash table collisions, and reduces slowpath forwarding of IP unicast and multicast traffic. For more information, see “Hardware Routing Table Management” on page 1262.

- To configure the number of IP routes to reserve in the LPM hardware table, use the command:

  ```
  configure iproute reserved-entries [ num_routes_needed | maximum | default ] slot [all | slot_num]
  ```

- To display the current configuration for IP route reserved entries, use the command:.

  ```
  show iproute reserved-entries {slot slot_num}
  ```

- To display the hardware table usage for IP routes, unicast and multicast, use the command:

  ```
  show iproute reserved-entries statistics {slot slot_num }
  ```

Configuring IP Route Sharing

IP route sharing is introduced in IP Route Sharing and ECMP on page 891. The following sections describe how to manage IP route sharing:

- Managing IP Route Sharing on page 909
- Viewing the IP Route Sharing Configuration on page 909

Managing IP Route Sharing

For SummitStack and ExtremeSwitching series switches that support Layer 3 routing, the ExtremeXOS software supports route sharing across up to 2, 4, 8, 16, 32, or 64 next-hop gateways.

- To configure the maximum number of *ECMP* gateways, use the following command:

  ```
  configure iproute sharing max-gateways max_gateways
  ```

  For guidelines on managing the number of gateways, see ECMP Hardware Table on page 905.

- To enable route sharing, use the following command:

  ```
  enable iproute {ipv4} sharing {{vr}vrname
  ```

- To disable route sharing, use the following command:

  ```
  disable iproute {ipv4} sharing {{vr} vrname
  ```

Viewing the IP Route Sharing Configuration

To view the route sharing configuration, use the following command:

```
show ipconfig {ipv4} {vlan vlan_name}
```

Configuring Route Compression

- To enable route compression for IPv4 routes, use the following command:

  `enable iproute compression {vr vrname}`

- To disable route compression for IPv4 routes, use the following command:

  `disable iproute compression {vr vrname}`

When you enable or disable route compression, that process is performed in chunks, rather than as one single processing event. Because the ExtremeXOS Route Manager processes a limited number of IP prefixes per second, route compression should not have any significant impact on performance. Likewise, when IP route compression is enabled, incremental route addition or deletion should not have a significant impact on performance.

Note
You should disable iproute compression if *MPLS* next hop is enabled on the switch.

Note
IP route compression is enabled by default.

Configuring Static Route Advertisement

- To enable or disable advertisement of all static routes, use one of the following commands:

  `enable rip export [bgp | direct | e-bgp | i-bgp | ospf | ospf-extern1 | ospf-extern2 | ospf-inter | ospf-intra | static | isis | isis-level-1| isis-level-1-external | isis-level-2 | isis-level-2-external ] [cost number {tag number} | policy policy-name]`

  or

  `disable rip export [bgp | direct | e-bgp | i-bgp | ospf | ospf-extern1 | ospf-extern2 | ospf-inter | ospf-intra | static | isis | isis-level-1| isis-level-1-external | isis-level-2| isis-level-2-external ]`

  `enable ospf export [bgp | direct | direct-inter-vr | e-bgp | i-bgp | rip | static | isis | isis-level-1 | isis-level-1-external | isis-level-2 | isis-level-2-external | host-mobility] [cost cost type [ase-type-1 | ase-type-2] {tag number} | policy-map]`

  or

  `disable ospf export [bgp | direct | direct-inter-vr | host-mobility |e-bgp | i-bgp | rip | static | isis | isis-level-1| isis-level-1-external | isis-level-2 | isis-level-2-external]`

Displaying the Routing Configuration and Statistics

Viewing IP Routes

Use the `show iproute` command to display the current configuration of IP unicast routing for the switch and for each *VLAN*. The `show iproute` command displays the currently configured routes and includes how each route was learned.

Viewing the IP ARP Table

To view the IP ARP table entries and configuration, use the `show iparp` command.

Viewing IP ARP Statistics

To view IP ARP table statistics, use the following commands:

`show iparp stats [[ vr_name | vr {all | vr_name} ] {no-refresh} | {vr} summary]`

`show iparp stats [vlan {all {vr vr_name}} | {vlan} vlan_name] {no-refresh}`

`show iparp stats ports {all | port_list} {no-refresh}`

Viewing the IP Configuration for a VLAN

To view the IP configuration for one or more *VLAN*s, use the `show ipconfig` command.

Viewing Compressed Routes

- View a compressed route using the following command: `show iproute`.

  Sample output:

```
Ori Destination        Gateway         Mtr  Flags       VLAN       Duration
#be 3.0.0.0/8          111.222.0.5     7    UG-D---um--- feed       0d:19h:52m:49s
#be 4.0.0.0/8          111.222.0.5     5    UG-D---um--- feed       0d:19h:52m:49s
#be 4.0.0.0/9          111.222.0.5     5    UG-D---um--c feed       0d:19h:52m:49s
#be 4.23.84.0/22       111.222.0.5     7    UG-D---um--c feed       0d:19h:52m:49s
#be 4.23.112.0/22      111.222.0.5     7    UG-D---um--c feed       0d:19h:52m:49s
.....................................
Origin(Ori): (b) BlackHole, (be) EBGP, (bg) BGP, (bi) IBGP, (bo) BOOTP
(ct) CBT, (d) Direct, (df) DownIF, (dv) DVMRP, (e1) ISISL1Ext
(e2) ISISL2Ext, (h) Hardcoded, (i) ICMP, (i1) ISISL1 (i2) ISISL2
(mb) MBGP, (mbe) MBGPExt, (mbi) MBGPInter
(mo) MOSPF (o) OSPF, (o1) OSPFExt1, (o2) OSPFExt2
(oa) OSPFIntra, (oe) OSPFAsExt, (or) OSPFInter, (pd) PIM-DM, (ps) PIM-SM
(r) RIP, (ra) RtAdvrt, (s) Static, (sv) SLB_VIP, (un) UnKnown
(*) Preferred unicast route (@) Preferred multicast route
(#) Preferred unicast and multicast route

Flags: (B) BlackHole, (D) Dynamic, (G) Gateway, (H) Host Route
(m) Multicast, (P) LPM-routing, (R) Modified, (S) Static
```

```
(u) Unicast, (U) Up (c) Compressed

Mask distribution:
19 routes at length  8           9 routes at length 9
9 routes at length 10          28 routes at length 11

Route Origin distribution:
7 routes from Direct          184816 routes from EBGP

Total number of routes = 184823
Total number of compressed routes = 93274
```

- Display an iproute summary using the following command: `show iproute summary`

  Sample output:

```
=================ROUTE SUMMARY=================
Mask distribution:
1 routes at length  8           7 routes at length 24
1 routes at length 32

Route origin distribution:
6    Static        3    Direct

Total number of routes = 9
Total number of compressed routes = 4
```

- Display a Route Manager configuration using the following command: `show configuration rtmgr`

  Sample output:

```
#
# Module rtmgr configuration.
#
disable iproute sharing
configure iproute priority mpls 20
.......
enable iproute compression ipv4 vr "VR-Default"
```

Routing Configuration Example

Figure 97 on page 913 illustrates a switch with three VLANs defined as follows:

- Finance
  - All ports on slots 1 and 3 have been assigned.
  - IP address 192.207.35.1.
- Personnel
  - Protocol-sensitive *VLAN* using the IP protocol.
  - All ports on slots 2 and 4 have been assigned.
  - IP address 192.207.36.1.
- MyCompany
  - Port-based VLAN.
  - All ports on slots 1 through 4 have been assigned.

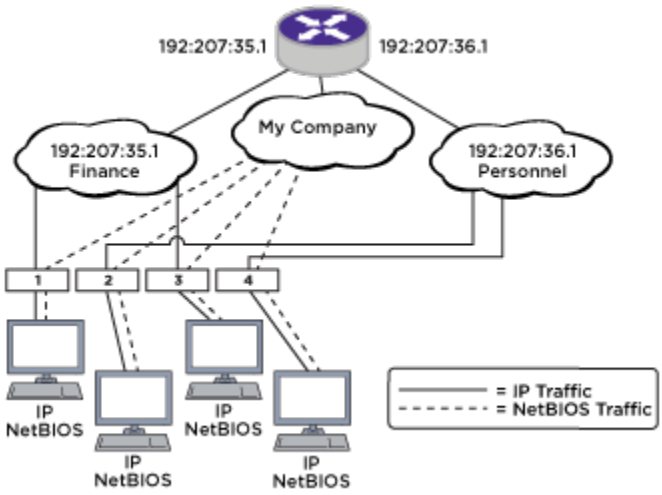


Figure 97: Unicast Routing Configuration Example

The stations connected to the system generate a combination of IP traffic and NetBIOS traffic. The IP traffic is filtered by the protocol-sensitive VLANs. All other traffic is directed to the VLAN MyCompany.

In this configuration, all IP traffic from stations connected to slots 1 and 3 have access to the router by way of the VLAN Finance. Ports on slots 2 and 4 reach the router by way of the VLAN Personnel. All other traffic (NetBIOS) is part of the VLAN MyCompany.

The example in the above figure is configured as follows:

```
create vlan Finance tag 10
create vlan Personnel tag 11
create vlan MyCompany

configure Finance protocol ip
configure Personnel protocol ip

configure Finance add port 1:*,3:* tag
configure Personnel add port 2:*,4:* tag
configure MyCompany add port all

configure Finance ipaddress 192.207.35.1
configure Personnel ipaddress 192.207.36.1
configure rip add vlan Finance
configure rip add vlan Personnel

enable ipforwarding
enable rip
```

Address Resolution Protocol (ARP) and Neighbor Discovery (ND)

For Address Resolution Protocol (ARP) and Neighbor Discovery (ND):

- The VR option in some configuration, clear, and show commands is not supported. Per VR table-based limits for ARP/ND maximum entries, maximum pending entries,

and maximum proxy entries are changed to a global single ARP/ND table-based limits. For more information, see the Switch Engine v33.6.1 Command References .

- ARP/ND entries may not be removed exactly when the timeout expires, since this dictated by the Linux behavior. They can be removed before or after than the configured timeout value.
- Any change in ARP timeout affects only newly learned entries. Existing entries are still bound to old timeout value.
- Starting with ExtremeXOS 30.1, the maximum configurable limit for IP ARP and ND maximum entries is changed to 157,696 and 49,152 respectively for all the switches. A message appears if the configured value exceeds the theoretical hardware maximum limit depending on the switch type.
- The output of `show configuration fdb` after upgrading to ExtremeXOS 30.1 is different from the 22.5 output because of the changes in per VR-based IPARP and ND configurations. For examples, see the Switch Engine v33.6.1 Command References .

Proxy ARP

Proxy Address Resolution Protocol (ARP) was first invented so that ARP-capable devices could respond to ARP request packets on behalf of ARP-incapable devices. Proxy ARP can also be used to achieve router redundancy and to simplify IP client configuration. The switch supports proxy ARP for this type of network configuration.

ARP-Incapable Devices

- Configure the switch to respond to ARP requests on behalf of devices that are incapable of doing so. Configure the IP address and MAC address of the ARP-incapable device using the following command: .

  ```
  configure iparp add proxy [ipNetmask | ip_addr {mask}] {vr vr_name}
  {mac | vrrp} {always}
  ```

After it is configured, the system responds to ARP requests on behalf of the device as long as the following conditions are satisfied:

- The valid IP ARP request is received on a router interface.
- The target IP address matches the IP address configured in the proxy ARP table.
- The proxy ARP table entry indicates that the system should answer this ARP request, based on the ingress *VLAN* and whether the always parameter is set. When the always option is set, the switch always responds to the ARP request even when the ARP requester is on the same subnet as the requested host. If the always option is not set, the switch only answers if the ARP request comes in from a VLAN that is not on the same subnet as the requested host.

When all the proxy ARP conditions are met, the switch formulates an ARP response using one of the following MAC addresses:

- A specific MAC address specified with the *mac* variable.
- The *VRRP (Virtual Router Redundancy Protocol)* virtual MAC address when the vrrp option is specified and the request is received on a VLAN that is running VRRP.

- The switch MAC address when neither of the above options applies.

Proxy ARP Support for Reachable Routes in the Network

You can configure whether the switch replies to *ARP (Address Resolution Protocol)* requests on a specified VLAN by proxy ARP if the route to the IP address is reachable, or only if proxy ARP entries have been created.

If an ARP request is received by the switch, it checks the ExtremeXOS proxy ARP table (user adds the entries through the CLI). If it is present, an ARP reply is sent. If not present, it searches for the entry in the kernel route table. If this IP address is reachable, then the ARP reply is sent. The default behavior is for the switch to reply to ARP requests on the specified VLAN(s) by proxy ARP only if proxy ARP entries have been created.

The following table summarizes the this feature's behavior:

	reachable	entry-required **(or command not configured)**
Entry present in proxy ARP Table (static entry added through command `configure iparp` **`add`** *`ip_addr`* {**`vr`** *`vr_name`*} *`mac`*)	Reply to ARP request	Reply to ARP request
Static entry not present in proxy ARP table	Reply to ARP request if route is reachable	No reply to ARP request
No static entry, but route reachable.	Reply to ARP request	
No static entry and route is not reachable	No reply to ARP request	
Static entry present and route is reachable	Reply to ARP request	
No static entry and route is not reachable	No reply to ARP request	
Route reachable	Reply to ARP request	
Route not reachable	No reply to ARP request	

Supported Platforms

All ExtremeSwitching Universal switches.

Limitations

- No support of SNMP.
- For this feature the statistics of ARP send “Out Response” is incremented.

To configure proxy ARP support for reachable routes, use the command:

```
configure iparp proxy [vlan all | {vlan} vlan_name] [reachable | entry-required]
```

To view the status for proxy ARP reachable routes on a VLAN, use the command:

```
show vlan {virtual-router vr-name}
```

Proxy ARP Between Subnets

In some networks, it is desirable to configure the IP host with a wider subnet than the actual subnet mask of the segment. You can use proxy ARP so that the router answers ARP requests for devices outside of the subnet. As a result, the host communicates as if all devices are local. In reality, communication with devices outside of the subnet are proxied by the router.

For example, an IP host is configured with a class B address of 100.101.102.103 and a mask of 255.255.0.0. The switch is configured with the IP address 100.101.102.1 and a mask of 255.255.255.0. The switch is also configured with a proxy ARP entry of IP address 100.101.0.0 and mask 255.255.0.0, without the always parameter.

When the IP host tries to communicate with the host at address 100.101.45.67, the IP host communicates as if the two hosts are on the same subnet, and sends out an IP ARP request. The switch answers on behalf of the device at address 100.101.45.67, using its own MAC address. All subsequent data packets from 100.101.102.103 are sent to the switch, and the switch routes the packets to 100.101.45.67.

IPv4 Multinetting

IP multinetting refers to having multiple IP networks on the same bridging domain (or *VLAN*). The hosts connected to the same physical segment can belong to any one of the networks, so multiple subnets can overlap onto the same physical segment. Any routing between the hosts in different networks is done through the router interface. Typically, different IP networks are on different physical segments, but IP multinetting does not require this.

Multinetting can be a critical element in a transition strategy, allowing a legacy assignment of IP addresses to coexist with newly configured hosts. However, because of the additional constraints introduced in troubleshooting and bandwidth, Extreme Networks recommends that you use multinetting as a transitional tactic only, and not as a long-term network design strategy.

Multinetting in ExtremeXOS does not require that you create a dummy multinetting protocol, and does not require that you create VLANs for each IP network. This implementation does not require you to explicitly enable IP multinetting. Multinetting is automatically enabled when a secondary IP address is assigned to a VLAN.

Multinetting Topology

For an IP multinetted interface, one of the IP networks on the interface acts as the transit network for the traffic that is routed by this interface. The transit network is

the primary subnet for the interface. The remaining multinetted subnets, called the secondary subnets, must be stub networks. This restriction is required because it is not possible to associate the source of the incoming routed traffic to a particular network. IP routing happens between the different subnets of the same *VLAN* (one arm routing) and also between subnets of different VLANs.

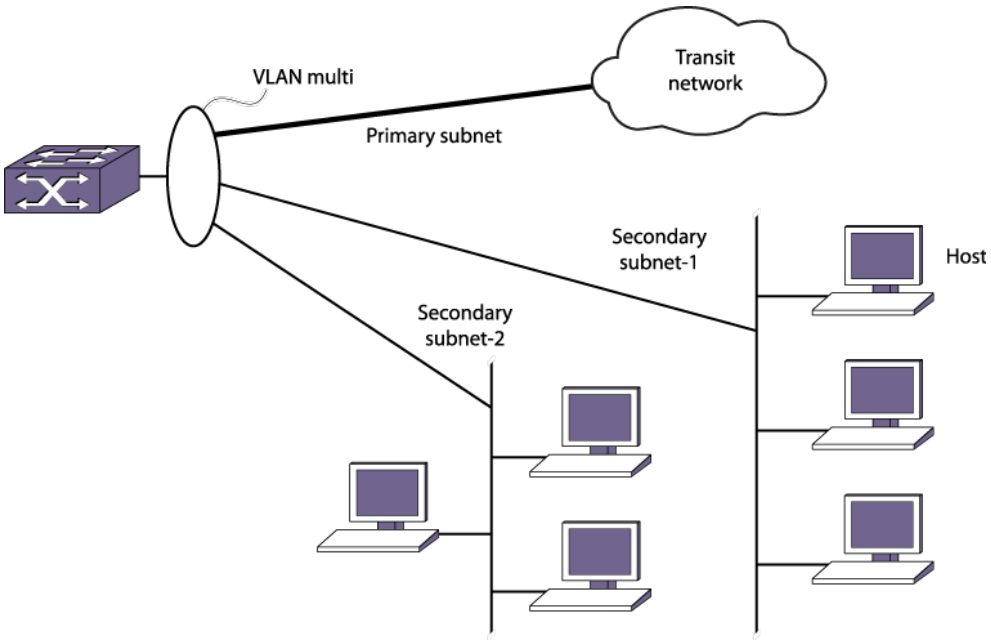


Figure 98: Multinetted Network Topology

Figure 210 shows a multinetted VLAN named multi. VLAN multi has three IP subnets so three IP addresses have been configured for the VLAN. One of the subnets is the primary subnet and can be connected to any transit network (for example, the Internet). The remaining two subnets are stub networks, and multiple hosts such as management stations (such as user PCs and file servers) can be connected to them. You should not put any additional routing or switching devices in the secondary subnets to avoid routing loops. In Figure 210 the subnets are on separate physical segments, however, multinetting can also support hosts from different IP subnets on the same physical segment.

When multinetting is configured on a VLAN, the switch can be reached using any of the subnet addresses (primary or secondary) assigned to VLAN. This means that you can perform operations like ping, Telnet, Trivial File Transfer Protocol (TFTP), Secure Shell 2 (SSH2), and others to the switch from a host residing in either the primary or the secondary subnet of the VLAN. Other host functions (such as traceroute) are also supported on the secondary interface of a VLAN.

How Multinetting Affects Other Features

ARP

Address Resolution Protocol is part of the TCP/IP suite used to dynamically associate a device's physical address (MAC address) with its logical address (IP address). The system broadcasts an ARP request, containing the IP address, and the device with that IP address sends back its MAC address so that traffic can be transmitted.

Route Manager

The Route Manager installs a route corresponding to each of the secondary interfaces. The route origin is direct, is treated as a regular IP route, and can be used for IP data traffic forwarding.

These routes can also be redistributed into the various routing protocol domains if you configure route redistribution.

IRDP

Some functional changes are required in Internet Router Discovery Protocol (IRDP) to support IP multinetting. When IRDP is enabled on a Layer 3 *VLAN*, the ExtremeXOS software periodically sends *ICMP* router advertisement messages through each subnet (primary and secondary) and responds to ICMP router solicitation messages based on the source IP address of the soliciting host.

Unicast Routing Protocols

Unicast routing protocols treat each IP network as an interface. The interface corresponding to the primary subnet is the active interface, and the interfaces corresponding to the secondary subnet are passive subnets.

For example, in the case of *OSPF*, the system treats each network as an interface, and hello messages are not sent out or received over the non-primary interface. In this way, the router link state advertisement (LSA) includes information to advertise that the primary network is a transit network and the secondary networks are stub networks, thereby preventing any traffic from being routed from a source in the secondary network.

Interface-based routing protocols (for example, OSPF) can be configured on per *VLAN* basis. A routing protocol cannot be configured on an individual primary or secondary interface. Configuring a protocol parameter on a VLAN automatically configures the parameter on all its associated primary and secondary interfaces. The same logic applies to configuring IP forwarding, for example, on a VLAN.

Routing protocols in the multinetted environment advertise the secondary subnets to their peers in their protocol exchange process. For example, for OSPF the secondary subnets are advertised as stub networks in router LSAs. *RIP* also advertises secondary subnets to its peers residing on the primary subnet.

This section describes the behavior of OSPF in an IPv4 multinetting environment:

- Each network is treated as an interface, and hello messages are not sent out or received over the non-primary interface. In this way, the router LSA includes information to advertise that the primary network is a transit network and the secondary networks are stub networks, thereby preventing any traffic from being routed from a source in the secondary network.
- Any inbound OSPF control packets from secondary interfaces are dropped.
- Direct routes corresponding to secondary interfaces can be exported into the OSPF domain (by enabling export of direct routes), if OSPF is not enabled on the container VLAN.

- When you create an OSPF area address range for aggregation, you must consider the secondary subnet addresses for any conflicts. That is, any secondary interface with the exact subnet address as the range cannot be in another area.
- The automatic selection algorithm for the OSPF router ID considers the secondary interface addresses also. The numerically highest interface address is selected as the OSPF router-id.

This section describes the behavior of the Routing Information Protocol (RIP) in an IP multinetting environment:

- RIP does not send any routing information update on the secondary interfaces. However, RIP does advertise networks corresponding to secondary interfaces in its routing information packet to the primary interface.
- Any inbound RIP control packets from secondary interfaces are dropped.
- Direct routes corresponding to secondary interfaces can be exported into the RIP domain (by enabling export of direct routes), if RIP is not enabled on the container VLAN.

There are no behavioral changes in the *BGP* in an IP multinetting environment.

This section describes a set of recommendations for using BGP with IP multinetting:

- Be careful of creating a BGP neighbor session with a BGP speaker residing in secondary subnet. This situation can lead to routing loops.
- All secondary subnets are like stub networks, so you must configure BGP in such a way that the BGP next hop becomes reachable using the primary subnet of a VLAN.
- When setting the BGP next hop using an inbound or outbound policy, ensure that the next hop is reachable from the primary interface.
- A BGP static network's reachability can also be resolved from the secondary subnet.
- Secondary interface addresses can be used as the source interface for a BGP neighbor.
- Direct routes corresponding to secondary interfaces can be exported into the BGP domain (by enabling export of direct routes).

This section describes the behavior of IS-IS in an IPv4 multinetting environment:

- IS-IS includes all the interface addresses in its reachability information. Adjacency is established only based on the primary interface address. If the adjacency-check option is disabled by the disable isis adjacency-check command, then IS-IS adjacency is established irrespective of the subnet address match.

IGMP Snooping and IGMP

IGMP (Internet Group Management Protocol) snooping and IGMP treat the *VLAN* as an interface. Only control packets with a source address belonging to the IP networks configured on that interface are accepted.

IGMP accepts membership information that originates from hosts in both the primary and secondary subnets. The following describes the changes in behavior of IGMP in an IP multinetting environment:

- A Layer 3 VLAN always uses the primary IP address as the source address to send out an IGMP query, and querier election is based on the primary IP address of interface. Because the RFC dictates that there is only one querier per physical segment, the querier may be attached to any of configured IP interfaces, including secondary interfaces, although this is not a recommended configuration.
- For a static IGMP group, the membership report is also sent out using the primary IP address.
- For local multicast groups such as 224.0.0.X, the membership report is sent out using the first IP address configured on the interface, which is the primary IP address in ExtremeXOS.
- The source IP address check is disabled for any IGMP control packets (such as IGMP query and IGMP membership report). Source IP address checking for IGMP control packet is disabled for all VLANs, not just the multinetted VLANs.

Multicast Routing Protocols

For Protocol-Independent Multicast (PIM), the following behavior changes should be noted in a multinetting environment:

- PIM does not peer with any other PIM router on a secondary subnet.
- PIM also processes data packets from the hosts secondary subnets.
- PIM also accepts membership information from hosts on secondary subnets.

EAPS, ESRP, and STP

Control protocols like *EAPS (Extreme Automatic Protection Switching)*, *ESRP (Extreme Standby Router Protocol)*, and the *STP (Spanning Tree Protocol)* treat the *VLAN* as an interface. If the protocol control packets are exchanged as Layer 3 packets, then the source address in the packet is validated against the IP networks configured on that interface.

DHCP Server

The *DHCP (Dynamic Host Configuration Protocol)* server implementation in ExtremeXOS only supports address allocation on the primary IP interface of the configured *VLAN*. That is, all DHCP clients residing on a bridging domain have IP addresses belonging to the primary subnet. To add a host on secondary subnet, you must manually configure the IP address information on that host.

DHCP Relay

When the switch is configured as a *DHCP* relay agent, it forwards the DHCP request received from a client to the DHCP server. When doing so, the system sets the GIADDR field in the DHCP request packet to the primary IP address of the ingress *VLAN*. This means that the DHCP server that resides on a remote subnet allocates an IP address for the client in the primary subnet range.

VRRP

This chapter assumes that you are already familiar with the Virtual Router Redundancy Protocol (VRRP). If not, refer to the following publications for additional information:

- RFC 2338—Virtual Router Redundancy Protocol (VRRP)
- RFC 2787—Definitions of Managed Objects for the Virtual Router Redundancy Protocol
- RFC 3768—Virtual Router Redundancy Protocol (VRRP) Version 2 for IPv4 www.ietf.org/rfc/rfc3768.txt
- RFC 5798—Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 http://tools.ietf.org/html/rfc5798
- Draft IETF VRRP Specification v2.06 http://tools.ietf.org/html/draft-ietf-vrrp-spec-v2-06

Configuring IPv4 Multinetting

You configure IP multinetting by adding a secondary IP address to a *VLAN*.

- Use the following command to add a secondary IP address:
  `configure vlan vlan_name add secondary-ipaddress [ip_address {netmask} | ipNetmask]`

  After you have added a secondary IP address, you cannot change the primary IP address of a VLAN until you first delete all the secondary IP addresses.
- To delete secondary IP addresses, use the following command:
  `configure vlan vlan_name delete secondary-ipaddress [ip_address | all]`

IP Multinetting Examples

The following example configures a switch to have one multinetted segment (port 5:5) that contains three subnets (192.168.34.0/24, 192.168.35.0/24, and 192.168.37.0/24).

Note
The secondary IP address of the super *VLAN* cannot be used for the sub VLAN IP address range.

```
configure default delete port 5:5
create vlan multinet
configure multinet ipaddress 192.168.34.1/24
configure multinet add secondary-ipaddress 192.168.35.1/24
configure multinet add secondary-ipaddress 192.168.37.1/24
configure multinet add port 5:5
enable ipforwarding
```

The following example configures a switch to have one multinetted segment (port 5:5) that contains three subnets (192.168.34.0, 192.168.35.0, and 192.168.37.0). It also configures a second multinetted segment consisting of two subnets (192.168.36.0 and 172.16.45.0). The second multinetted segment spans three ports (1:8, 2:9, and 3:10). *RIP* is enabled on both multinetted segments.

```
configure default delete port 5:5
create vlan multinet
```

```
configure multinet ipaddress 192.168.34.1
configure multinet add secondary-ipaddress 192.168.35.1
configure multinet add secondary-ipaddress 192.168.37.1
configure multinet add port 5:5
configure default delete port 1:8, 2:9, 3:10
create vlan multinet_2
configure multinet_2 ipaddress 192.168.36.1
configure multinet_2 add secondary-ipaddress 172.16.45.1
configure multinet_2 add port 1:8, 2:9, 3:10
configure rip add vlan multinet
configure rip add vlan multinet_2
enable rip
enable ipforwarding
```

DHCP/BOOTP Relay

The following sections describe how to use the *DHCP*/BOOTP Relay feature:

- Managing DHCP/BOOTP Relay on page 922.
- Configuring the DHCP Relay Agent Option (Option 82) at Layer 3 on page 923.
- Viewing the DHCP/BOOTP Relay Statistics and Configuration on page 925.

Note
This section discusses DHCP/BOOTP relay operation at Layer 3. For information on DHCP/BOOTP relay operation at Layer 2, see DHCP Snooping and Trusted DHCP Server on page 1565.

BOOTP Relay agent of DHCPv6 relays the DHCPv6 messages between the server/client across subnets of a larger IPv6 network.

The DHCPv6 server/BOOTP relay agent listens to UDP port 547.

A relay agent relays both messages from clients and Relay-forward messages from other relay agents. When a relay agent receives a valid message, it constructs a new Relay-forward message and option from the DHCP message received, then forwards it to the next hop/server. The server responds with the corresponding IP address or configuration through a Relay-Reply message to its peer, and thus to the client.

The ExtremeXOS implementation of DHCPv6 takes reference from ISC DHCPv6.

Managing DHCP/BOOTP Relay

After IP unicast routing has been configured, you can configure the switch to forward *DHCP* or BOOTP requests coming from clients on subnets being serviced by the switch and going to hosts on different subnets. This feature can be used in various applications, including DHCP services between Windows NT servers and clients running Windows 95.

Note
If DHCP/BOOTP Relay is enabled on a per *VLAN* basis, make sure it is enabled on both the client-side and server-side VLANs.

You can enable BOOTP Relay over VRFs and VPN-VRFs in the same manner as with normal VRs. For more information about BOOTP Relay over L3VPN, see L3VPN BOOTP Relay on page 1030.

Note
BOOTP Relay show commands do not increment DHCP Release counters that are received from the client since DHCP release is hardware forwarded. Counters are incremented for other options in these commands.

Configuring the DHCP Relay Agent Option (Option 82) at Layer 3

After configuring and enabling the *DHCP*/BOOTP relay feature, you can enable the DHCP relay agent option feature. This feature inserts a piece of information, called option 82, into any DHCP request packet that is to be relayed by the switch. Similarly, if a DHCP reply received by the switch contains a valid relay agent option, the option is stripped from the packet before it is relayed to the client.

When DHCP option 82 is enabled, two types of packets need to be handled:

DHCP Request

When the switch (relay agent) receives a DHCP request, option 82 is added at the end of the packet. If the option has already been enabled, then the action taken depends on the configured policy (drop packet, keep existing option 82 value, or replace the existing option). If the incoming DHCP request is tagged, then that *VLAN* ID is added to the circuit ID sub option of option 82; otherwise, the default VLAN ID is added.

DHCP Reply

When the option 82 information check is enabled, the packets received from the DHCP server are checked for option 82 information. If the remote ID sub-option is the switch's MAC address, the packet is sent to the client; if not, the packet is dropped. If the check is not enabled. The packets are forwarded as-is.

The DHCP relay agent option consists of two pieces of data, called sub-options. The first is the agent circuit ID sub-option, and the second is the agent remote ID sub-option. When the DHCP relay agent option is enabled on switches running ExtremeXOS, the value of these sub-options is set as follows:

Agent circuit ID sub-option

The full circuit ID string uses the format <vlan_info>-<port_info>. You can use the default values for vlan_info and port_info, or you can configure these values as described later in this section.

Agent remote ID sub-option

Always contains the Ethernet MAC address of the relaying switch. You can display the Ethernet MAC address of the switch by issuing the show switch command. You can configure string up to 32 characters as agent remote ID option.

Note

For more general information about the DHCP relay agent information option, refer to RFC 3046.

Enabling and Disabling the DHCP Relay Agent Option

- To enable the *DHCP* relay agent option, use the following command after configuring the DHCP/BOOTP relay function:

  ```
  configure bootprelay dhcp-agent information option
  ```

- To disable the DHCP relay agent option, use the following command:

  ```
  unconfigure bootprelay dhcp-agent information option
  ```

Enabling and Disabling DHCP Packet Checking

In some instances, a *DHCP* server may not properly handle a DHCP request packet containing a relay agent option.

- To prevent DHCP reply packets with invalid or missing relay agent options from being forwarded to the client, use this command:

  ```
  configure bootprelay dhcp-agent information check
  ```

- To disable checking of DHCP replies, use this command:

  ```
  unconfigure bootprelay dhcp-agent information check
  ```

Configuring the DHCP Packet Handling Policy

A *DHCP* relay agent may receive a client DHCP packet that has been forwarded from another relay agent. If this relayed packet already contains a relay agent option, then the switch handles this packet according to the configured DHCP relay agent option policy. The possible actions are to replace the option information, to keep the information, or to drop packets containing option 82 information.

- To configure this policy, use the following command:

  ```
  configure bootprelay dhcp-agent information policy [drop | keep |
  replace]
  ```

- The default relay policy is replace. Configure the policy to the default, use this command:

  ```
  unconfigure bootprelay dhcp-agent information policy
  ```

Configuring the DHCP Agent Circuit ID Suboption

- To configure the values used to create the agent circuit ID suboption, use the following commands:

  ```
  configure bootprelay dhcp-agent information circuit-id port-
  information port_info port port
  ```

```
configure bootprelay dhcp-agent information circuit-id vlan-
information vlan_info {vlan} [vlan_name |all]
```

Note
When both ipsecurity DHCP option and bootprelay DHCP agent option are configured, the Bootprelay DHCP agent option takes higher precedence.

Viewing the DHCP/BOOTP Relay Statistics and Configuration

- To view the *DHCP*/BOOTP relay statistics and configuration, use the following command:

  ```
  show bootprelay
  ```

- To view the BOOTP relay enable/disable configuration, use the following command:

  ```
  show bootprelay configuration {{vlan} vlan_name | {vr} vr_name}
  ```

- To view the DHCP relay agent option (Option 82) configuration, use the following commands:

  ```
  show bootprelay dhcp-agent information circuit-id port-information
  ports all
  ```

  ```
  show bootprelay dhcp-agent information circuit-id vlan-information
  ```

DHCP Smart Relay

A *DHCP* Relay agent relays DHCP requests from the client to the DHCP server, and relays the DHCP replies from the server to the client. It acts as a proxy, and can reduce the number of DHCP servers required in the network. The DHCP relay agent inserts its own IP address in the giaddr field (gateway address) of the DHCP request. The DHCP server looks into this IP address, identifys the DHCP client's subnet, and assigns an IP address accordingly.

The ExtremeXOS BOOTPRelay module is Extreme Networks' DHCP Relay agent. It is now enhanced to optionally insert the secondary addresses of the interface.

DHCP clients can now be dynamically assigned both public and private addresses in an effort to reduce administrative overhead. ExtremeXOS can now configure the BOOTPRelay agent to insert both primary and secondary address(es) of the client-facing interface as a gateway address in the DHCP packet. At any given point in time, the DHCP client can be assigned one IP address.

You can insert different addresses in the giaddr field (gateway address) in two different ways:

- Parallel mode: the switch simultaneously relays multiple DHCP Discover packets, each containing a different IP address as the gateway address. The relay agent receives a DHCP Discover request from the DHCP client. The relay agent makes multiple copies of this DHCP DISCOVER request, inserts each IP address of the client-facing interface in the giaddr field (gateway address) in each one of these copies, and relays all these DHCP DISCOVER packets simultaneously to the DHCP server.

The DHCP server has the responsibility to assign the correct IP address in the correct subnet by choosing the DHCP DISCOVER packet to respond to. Similarly, the DHCP client is responsible for accepting the appropriate DHCP OFFER from the DHCP server.

Note
Only the DHCP DISCOVER request is sent in multiple copies, with different IP addresses as the gateway address in each. All other DHCP packets are relayed normally.

- Sequential mode: the switch relays a DHCP DISCOVER packet for each IP address sequentially.

  The relay agent receives a DHCP DISCOVER request and inserts the primary address of the client-facing interface as the gateway address, and relays this packet to the server. The switch counts the number of times a DHCP client sends out a DHCP request without receiving a DHCP OFFER message. After three retries, the relay agents sets the secondary address as the gateway address in the next DHCP Discover request that gets relayed. If the DHCP server still does not respond after three retries, the next secondary address is used as the gateway address, and so on cyclically.

Configuring DHCP Smart Relay

Issue the following commands to configure te DHCP Smart Relay feature:

- To configure DHCP smart relay mode to include the secondary IP address as giaddr at VR level: `configure bootprelay {ipv4 | ipv6} include-secondary {sequential | parallel | off} {vr vr_name}`
- To configure DHCP smart relay mode to include the secondary IP address as giaddr at *VLAN* level: `configure bootprelay {ipv4 | ipv6} {vlan} vlan_name include-secondary {sequential | parallel | off}`
- To unconfigure any smart relay configuration that was specified at the VLAN level: `unconfigure bootprelay {ipv4 | ipv6} {vlan} vlan_name include-secondary`
- To display various DHCP BOOTP Relay statistics:
  - `show bootprelay`
  - `show bootprelay ipv6`
  - `show bootprelay configuration ipv4`
  - `show bootprelay configuration ipv6`

Supported Platforms

This feature is supported on all platforms.

Broadcast UDP Packet Forwarding

UDP Forwarding is a flexible and generalized routing utility for handling the directed forwarding of broadcast UDP packets. UDP Forwarding enables you to configure your switch so that inbound broadcast UDP packets on a *VLAN* are forwarded to a particular destination IP address or VLAN. UDP Forwarding allows applications, such as multiple *DHCP* relay services from differing sets of VLANs, to be directed to different DHCP servers.

The following rules apply to UDP broadcast packets handled by this feature:

- If the UDP profile includes BOOTP or DHCP, it is handled according to guidelines specified in RFC 1542.
- If the UDP profile includes other types of traffic, these packets have the IP destination address modified as configured, and changes are made to the IP and UDP checksums and TTL field (decrements), as appropriate.

If UDP Forwarding is used for BOOTP or DHCP forwarding purposes, do not configure or use the existing bootprelay function. However, if the previous bootprelay functions are adequate, you may continue to use them.

Note

When using udp-profile to forward dhcp request, the behavior will be different from bootprelay. Where bootprelay will forward the dhcp packet with the client vlan IP as source IP, udp-profile will forward the dhcp packet with the source IP of the egress vlan towards the destination server.

UDP Forwarding only works across a Layer 3 boundary and currently, UDP Forwarding can be applied to IPv4 packets only, not to IPv6 packets.

Note

In ExtremeXOS 22.6 and earlier, IPv4 UDP broadcast packets are forwarded to the destination IP address as configured in UDP policy without performing `enable ipforwarding` on the action VLAN. Starting with ExtremeXOS 30.1, IPv4 UDP broadcast packets are forwarded to destination IP address as configured in UDP policy only if `enable ipforwarding` is enabled on the action VLAN.

Configuring UDP Forwarding

- To configure UDP Forwarding, create a policy file for your UDP profile, and then associate the profile with a *VLAN* using the following command:

  `configure vlan` *`vlan_name`* **`udp-profile`** [*`profilename`* | **`none`**]

  You can apply a UDP Forwarding policy only to an L3 VLAN (a VLAN having at least one IP address configured on it). If no IP address is configured on the VLAN, the command is rejected.

UDP profiles are similar to *ACL (Access Control List)* policy files.UDP profiles use a subset of the match conditions allowed for ACLs. Unrecognized attributes are ignored. A UDP forwarding policy must contain only the following attributes:

- Match attributes
  - Destination UDP port number (destination-port)
  - Source IP address (source-ipaddress)
- Action modified (set) attributes
  - Destination IP address (destination-ipaddress)
  - VLAN name (vlan)

Policy files used for UDP forwarding are processed differently from standard policy files. Instead of terminating when an entry's match clause becomes true, each entry in the policy file is processed and the corresponding action is taken for each true match clause.

For example, if the following policy file is used as a UDP forwarding profile, any packets destined for UDP port 67 are sent to IP address 20.0.0.5 and flooded to VLAN to7:

```
entry one {
if match all {
    destination-port 67 ;
} then {
    destination-ipaddress 20.0.0.5 ;
}
}
entry two {
if match all {
    destination-port 67 ;
} then {
    vlan "to7" ;
}
}
```

If you include more than one VLAN set attribute or more than one destination-ipaddress set attribute in one policy entry, the last one is accepted and the rest are ignored.

Note
Although the XOS policy manager allows you to set a range for the destination-port, you should not specify the range for the destination-port attribute in the match clause of the policy statement for the UDP profile. If a destination-port range is configured, the last port in the range is accepted and the rest are ignored.

You can have two valid set statements in each entry of a UDP forwarding policy; one a destination-ipaddress and one a VLAN. The ExtremeXOS software currently allows a maximum of eight entries in a UDP forwarding policy, so you can define

a maximum of 16 destinations for one inbound broadcast UDP packet: eight IP addresses and eight VLANs.

Note

It is strongly advised to have no more than eight entries in a UDP forwarding profile. The UDP forwarding module processes those entries even if the entries do not contain any attributes for UDP forwarding. Having more than eight entries drastically reduces the performance of the system. If the inbound UDP traffic rate is very high, having more than eight entries could cause the system to freeze or become locked.

If you rename a VLAN referred to in your UDP forwarding profile, you must manually edit the policy to reflect the new name, and refresh the policy.

You can also validate whether the UDP profile has been successfully associated with the VLAN by using the show policy command. UDP Forwarding is implemented as part of the netTools process, so the command does display netTools as a user of the policy.

- To remove a policy, use the none form of the following command:

  `configure vlan` *`vlan_name`* **`udp-profile`** [*`profilename`* | **`none`**]

  or use the following command:

  `unconfigure vlan` *`vlan_name`* **`udp-profile`**

For more information about creating and editing policy files, see Chapter 17, “Policy Manager.” For more information about ACL policy files, see Chapter 18, “ACLs.”

Configuring UDP Echo Server Support

You can use UDP echo packets to measure the transit time for data between the transmitting and receiving ends.

- To enable UDP echo server support, use the following command:

  `enable udp-echo-server {`**`vr`** *`vrid`*`}{`**`udp-port`** *`port`*`}`

- To disable UDP echo server support, use the following command:

  `disable udp-echo-server {`**`vr`** *`vrid`*`}`

IP Broadcast Handling

The ExtremeXOS software supports IP subnet directed broadcast forwarding. In the ExtremeXOS software, IP subnet directed broadcast forwarding is done in the software by default; if you want to perform forwarding in the hardware, see the command reference pages on IP forwarding in the Switch Engine v33.6.1 Command References .

IP Broadcast Handling Overview

To understand how IP broadcast handling functions in the ExtremeXOS software, consider the following two examples.

For the first example, a system sends an IP packet (such as the IP packet generated by the ping command) to an IP subnet directed broadcast address which is directly connected to that system. In this case, the IP packet goes out as a Layer 2 broadcast with the destination media access control (DMAC) addresses all set to FF, while the source media access control (SMAC) is set to the system MAC. This packet is sent out of all the ports of the *VLAN*.

In the second example, a system sends a packet (such as the IP packet generated by the ping command) to an IP subnet directed broadcast address which is remotely connected through a gateway. In this case, the IP packet goes out as a Layer 2 unicast packet with the DMAC equal to the gateway's MAC address, while the SMAC is set to the system MAC. At the gateway router, the existing IP packet forwarding mechanism is sufficient to send the packet out of the correct interface if the router is not the final hop router.

When the packet reaches the final hop router, which is directly connected to the target IP subnet, IP directed broadcast forwarding needs to be turned on.

The IP broadcast handling feature is applicable only at the final hop router directly attached to the target subnet. At the final hop router, when IP subnet directed broadcast forwarding is enabled on an IP VLAN via the command line, the following happens:

- Some basic validity checks are performed (for example, a check to see if the VLAN has IP enabled)
- A subnet broadcast route entry for the subnet is installed. For example, consider a system with the following configuration:

  VLAN-A = 10.1.1.0/24, ports 1:1, 1:2, 1:3, 1:4

  VLAN-B = 20.1.1.0/24, ports 1:5, 1:6, 1:7, 1:8

  VLAN-C = 30.1.1.0/24, ports 1:9, 1:10, 1:11

  If you enable IP directed broadcast forwarding on VLAN-A, you should install a route entry for 10.1.1.255 on this system.
- A packet arriving on port 1:5 VLAN-B with destination IP (DIP) set to 10.1.1.255, the source IP (SIP) set to 20.1.1.3, the DMAC set to the router MAC, and the SMAC set to the originating system MAC, arrives at the installed route entry and is sent out on all the ports of VLAN-A, with DMAC set to be all FF and the SMAC set to the router's system MAC.
- An IP packet arriving on port 1:1 VLAN-A with the DIP set to 10.1.1.255, the SIP set to 10.1.1.3, the DMAC set to all FF, and the SMAC set to the originator’s MAC, causes Layer 2 flooding on all ports of VLAN-A.

When IP subnet directed broadcast is disabled on an IP VLAN, it is disabled on all VLAN ports and all IP subnet directed broadcast entries are deleted.

Note
IP subnet directed broadcast uses fast-path forwarding.

VLAN Aggregation

VLAN aggregation is a feature aimed primarily at service providers.

Note
This feature is supported only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

The purpose of VLAN aggregation is to increase the efficiency of IP address space usage. It does this by allowing clients within the same IP subnet to use different broadcast domains while still using the same default router.

Using VLAN aggregation, a superVLAN is defined with the desired IP address. The subVLANs use the IP address of the superVLAN as the default router address. Groups of clients are then assigned to subVLANs that have no IP address, but are members of the superVLAN. In addition, clients can be informally allocated any valid IP addresses within the subnet. Optionally, you can prevent communication between subVLANs for isolation purposes. As a result, subVLANs can be quite small, but allow for growth without re-defining subnet boundaries.

Without using VLAN aggregation, each VLAN has a default router address, and you need to use large subnet masks. The result of this is more unused IP address space.

Multiple secondary IP addresses can be assigned to the superVLAN.

Figure 99 illustrates VLAN aggregation.

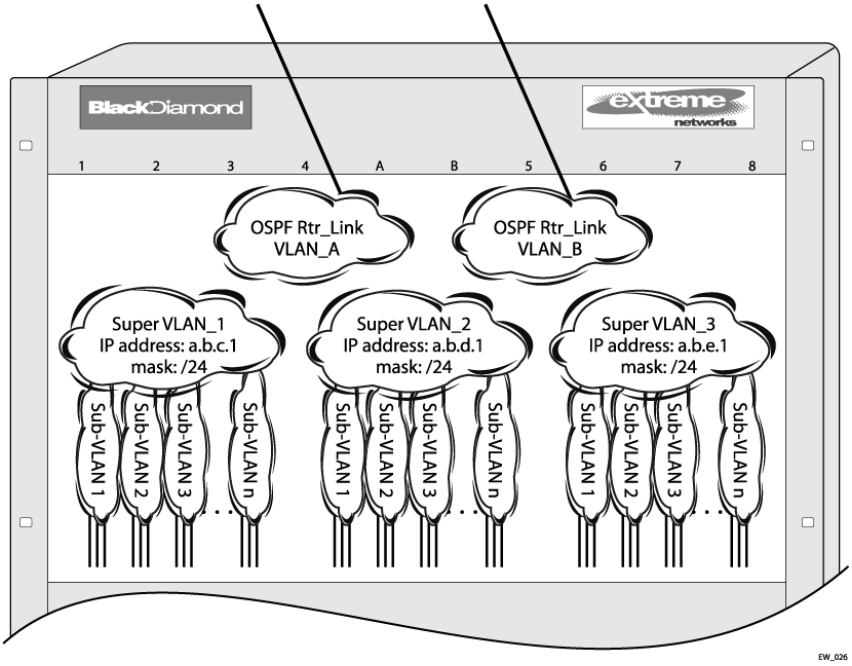


Figure 99: VLAN Aggregation

In this example, all stations are configured to use the address 10.3.2.1 for the default router.

VLAN Aggregation Properties

VLAN aggregation is a very specific application, and the following properties apply to its operation:

- All broadcast and unknown traffic remains local to the subVLAN and does not cross the subVLAN boundary. All traffic within the subVLAN is switched by the subVLAN, allowing traffic separation between subVLANs (while using the same default router address among the subVLANs).
- Hosts can be located on the superVLAN or on subVLANs. Each host can assume any IP address within the address range of the superVLAN router interface. Hosts on the subVLAN are expected to have the same network mask as the superVLAN and have their default router set to the IP address of the superVLAN.
- All IP unicast traffic between subVLANs is routed through the superVLAN. For example, no *ICMP* redirects are generated for traffic between subVLANs, because the superVLAN is responsible for subVLAN routing. Unicast IP traffic across the subVLANs is facilitated by the automatic addition of an ARP entry (similar to a proxy ARP entry) when a subVLAN is added to a superVLAN. This feature can be disabled for security purposes.

VLAN Aggregation Limitations

The following limitations apply to *VLAN* aggregation:

- No additional routers may be located in a subVLAN. This feature is only applicable for “leaves” of a network.
- A subVLAN cannot be a superVLAN, and vice-versa.
- SubVLANs are not assigned an IP address.
- A subVLAN should belong to only one superVLAN.
- A subVLAN or superVLAN should not be added to a private VLAN
- Before you can delete a superVLAN, you must delete all subVLANs in that superVLAN.
- When configuring a subVLAN address range, all addresses in the range must belong to the superVLAN subnet.

SubVLAN Address Range Checking

You can configure subVLAN address ranges on each subVLAN to prohibit the entry of IP addresses from hosts outside of the configured range.

- To configure a subVLAN range, use the following command:

  `configure vlan` *`vlan_name`* **`subvlan-address-range`** *`ip address1 - ip address2`*

- To remove a subVLAN address range, use the following command:

  `unconfigure vlan` *`vlan_name`* **`subvlan-address-range`**

- To view the subVLAN address range, use the following command:

  ```
  show dhcp-server {vlan vlan_name}
  ```

  Note
  Also, for super VLANs/SubVLAN, to assign an IP address, any trunk port added to a super VLAN needs to be enabled for DHCP server.

Isolation Option for Communication Between SubVLANs

To facilitate communication between sub*VLAN*s, by default, an entry is made in the IP ARP table of the superVLAN that performs a proxy ARP function. This allows clients on one subVLAN to communicate with clients on another subVLAN. In certain circumstances, intra-subVLAN communication may not be desired for isolation reasons.

- To prevent normal communication between subVLANs, disable the automatic addition of the IP ARP entries on the superVLAN using the following command:

  ```
  disable subvlan-proxy-arp vlan [vlan-name | all]
  ```

  Note
  The isolation option works for normal, dynamic, ARP-based client communication.

VLAN Aggregation Example

The follow example illustrates how to configure *VLAN* aggregation. The VLAN vsuper is created as a superVLAN, and subVLANs, vsub1, vsub2, and vsub3 are added to it.

1. Create and assign an IP address to a VLAN designated as the superVLAN. Be sure to enable IP forwarding and any desired routing protocol on the switch.

   ```
   create vlan vsuper
   configure vsuper ipaddress 192.201.3.1/24
   enable ipforwarding
   enable ospf
   configure ospf add vsuper area 0
   ```

2. Create and add ports to the subVLANs.

   ```
   create vlan vsub1
   configure vsub1 add port 10-12
   create vlan vsub2
   configure vsub2 add port 13-15
   create vlan vsub3
   configure vsub3 add port 16-18
   ```

3. Configure the superVLAN by adding the subVLANs.

   ```
   configure vsuper add subvlan vsub1
   configure vsuper add subvlan vsub2
   configure vsuper add subvlan vsub3
   ```

4. Optionally, disable communication among subVLANs.

```
disable subvlan-proxy-arp vlan all
```

Note
The above command has no impact on Layer 3 traffic.

Verify the VLAN Aggregation Configuration

The following commands can be used to verify proper *VLAN* aggregation configuration:

- `show vlan`—Indicates the membership of subVLANs in a superVLAN.
- `show iparp`—Indicates an ARP entry that contains subVLAN information. Communication with a client on a subVLAN must have occurred in order for an entry to be made in the ARP table.

IP Network Address Translation (NAT)

Network Address Translation (NAT) is a method by which IP addresses are mapped from one address domain or realm to another, in an attempt to provide transparent routing to hosts. Traditionally, NAT devices are used to connect an isolated address realm with private unregistered addresses to an external realm with globally unique registered addresses.

Traditional NAT allows hosts within a private network to transparently access hosts in the external network. In a traditional NAT, sessions are uni-directional. Internal network hosts typically initiate sessions to external servers using public addresses. When hosts in a private network initiate a session to an external server, the NAT router detects packets as they cross realms and installs address mappings for the session. After the session is established, the two hosts can communicate transparently with the help of NAT. There are three variations to traditional NAT:

- Basic static source NAT
- Dynamic source Network Address Port Translation (NAPT)
- Static destination NAPT

Note
See *Feature License Requirements* for supported platforms.

Basic Static Source NAT

In basic source NAT, the NAT router translates and replaces the IP address in the packet IP header.

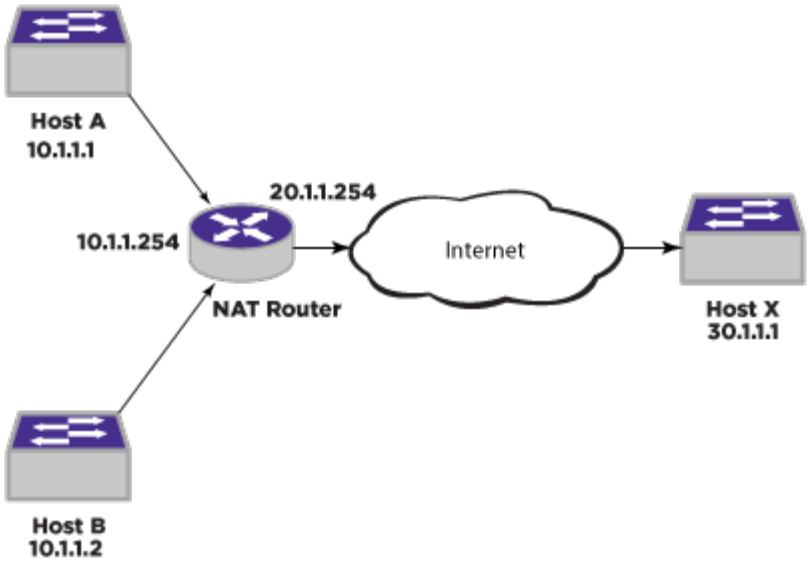


Figure 100: Example Network for NAT and NAPT

In Figure 100, there are two address domains. The address domain 10.1.1.0/24 on the NAT router is an internal IP domain, and the address domain 20.1.1.0/24 is an external (public) domain. The NAT router has a pool of valid global IP addresses, one for each internal host that it expects to initiate sessions with external servers. The NAT router maintains a NAT table to map the internal IP address to the global IP address and vice versa. The administrator must configure the mapping of each internal host address to the external host IP address. For example, the administrator can configure internal IP addresses 10.1.1.1 and 10.1.1.2 to map to external IP address as 20.1.1.100 and 20.1.1.101, respectively. When hosts 10.1.1.1 and 10.1.1.2 want to communicate with an Internet host whose IP address is 30.1.1.1, the translations as shown in Figure 101 occur.

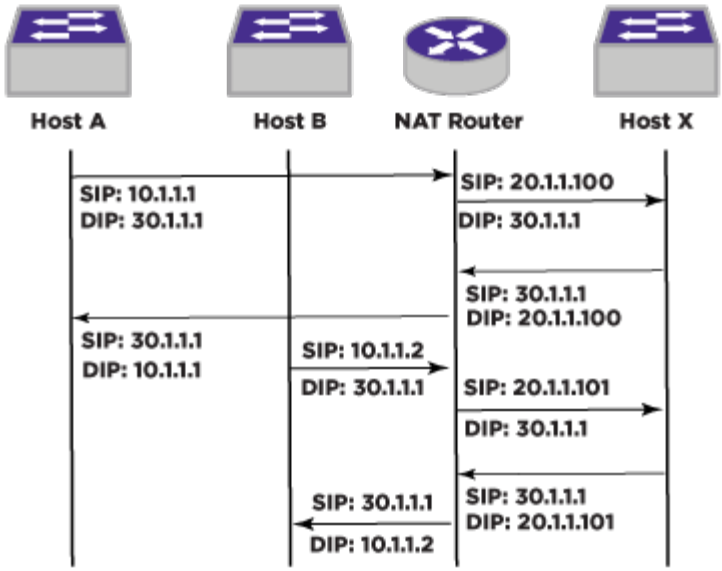


Figure 101: Address Translations During Basic NAT

In the outbound direction, the NAT router performs SNAT (source NAT), and replaces the source IP address (SIP) of the packet. In the inbound direction, the NAT router performs DNAT (destination NAT), and replaces the destination IP address (DIP) of the packet. If the number of registered public IP addresses available to the NAT router is equal to or greater than the number of internal hosts, this mode of translation can be used. If the number of public IP addresses is less than the number of internal hosts, NAPT is used.

Do not use the primary IP address of the “out” VLAN for translation. The NAT router cannot differentiate if the traffic has to be translated and routed, or if it is destined to the NAT router and has to be processed.

Add the translation IP as a secondary address to the “out” VLAN. This secondary IP should never be used by any other protocols like BGP or OSPF. The translated IP address for each rule has to be unique for source NAT.

The translated IP can be configured as a secondary IP address or any IP address that is not "owned" by the NAT router. If the IP address is not "owned," then a proxy ARP entry should be added for this IP address so that the NAT router responds to the ARP requests.

Each VLAN can have 254 secondary IP addresses. If secondary IP addresses are used for translation, the number of source NAT rules is limited to 254 times the number of egress NAT VLANs.

Source Network Address Port Translation

Source Network Address Port Translation (NAPT) translates the transport identifier (for example, TCP and UDP port numbers or ICMP query identifiers) in addition to the IP

addresses. This translation mechanism allows multiple hosts in the private network to share a single external address.

In the example shown in Figure 100 on page 935, there are two address domains. The address domain 10.1.1.0/24 on the NAT router is an internal IP domain and the address domain 20.1.1.0/24 is an external (public) domain. The NAT router has only one valid Internet IP address for all the internal hosts to share. The NAT router maintains a NAT table that maps {Internal IP, IP protocol, L4 port} tuple to the the {global IP, IP protocol, modified L4 port} tuple, and vice versa.

The administrator has to configure a base NAPT rule giving the range of input IP addresses to be translated and the external IP to which these have to be translated. For each of the new flows that do not have a NAT rule setup in the hardware, a “NAT miss” is generated and the packets are lifted to the CPU. The application picks up an unused port number and programs the hardware with a dynamic NAT rule. After this, NAT translations happen in hardware.

Figure 102 depicts the translations that happen with NAPT for two hosts with IP addresses of 10.1.1.1 and 10.1.1.2. Host A is sending TCP packets destined to TCP port 1000 on Host X (30.1.1.1). Host B is sending UDP packets destined to UDP port 2000 on Host X (30.1.1.1).

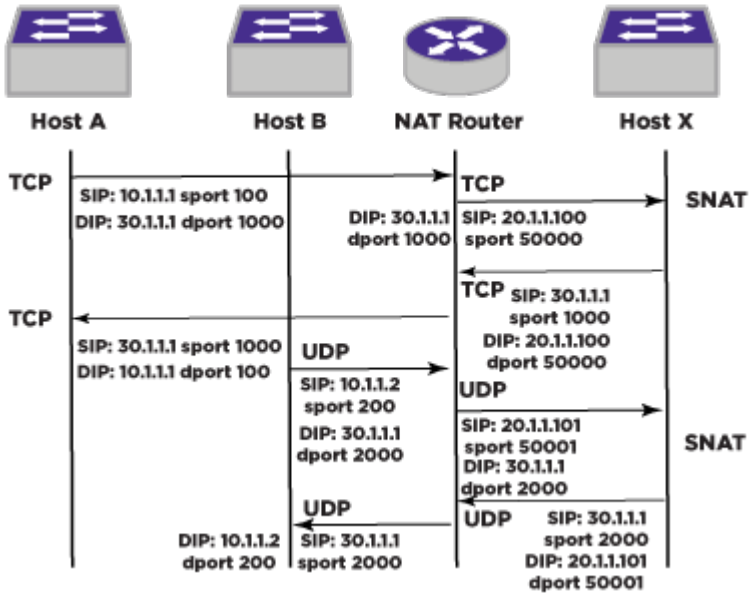


Figure 102: NAPT Example

In the outbound direction, the NAT router performs SNAT (source NAT), and replaces the source IP address (SIP) and source port (sport) of the packet. In the inbound direction, the NAT router performs DNAT (destination NAT) and replaces the destination IP address (DIP) and the destination port (dport) of the packet. Note that the source IP address of the outgoing packet for the streams from both hosts remains the same here. NAPT works only for traffic with a suitable Transport Layer identifier: specifically, TCP, UDP, and ICMP traffic. NAPT does not work with fragmented IP packets and are dropped when NAT is enabled. This is because the fragmented IP packet does not contain a valid TCP/UDP header unless it is the first fragment.

Software Aging of NAT Entries

Software aging is performed for dynamically created entries. The entries that are not refreshed for the configured age interval are deleted. The default value of the age interval is 20 minutes.

Destination Network Address Port Translation

Destination Network Address Port Translation (also known as Port forwarding) allows remote computers to initiate connections to specific servers within the private network. The private IP addresses of the servers within the private network cannot be used by the remote computers to directly access them. To access the internal servers, the remote computers typically connect to the NAT router on a configured port number. The NAT router modifies the destination IP address and the port number in the inward direction before sending it to the internal server. In the outward direction, it changes the source IP address and port number before sending it back to the remote computer.

In Figure 100 on page 935, if Host X (30.1.1.1) wants to access the HTTP service on Host B (10.1.1.2), it has to send the traffic through the NAT router. A port is configured on the NAT router to redirect the traffic towards the internal host. Remote Host X has to use the IP address of NAT router along with this configured port number.

If the translation port is configured as 8080, the translations that happen in the packet are shown in Figure 103.

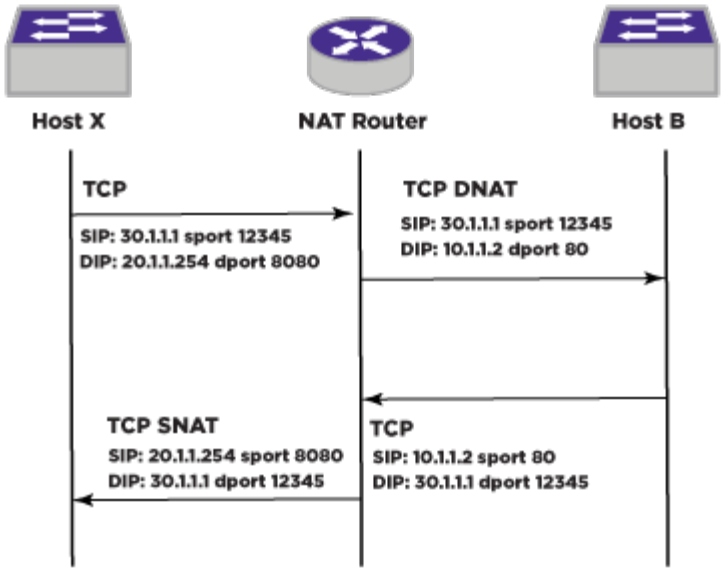


Figure 103: Destination Network Address Port Translation (NAPT)

Configuring IP Network Address Translation (NAT)

To enable or disable NAT globally, use the following commands:

```
enable ip nat
```

```
disable ip nat
```

To configure NAT VLANs, use the following commands:

```
configure ip nat add {vlan} vlan_name direction [ingress | egress |
both]
```

```
configure ip nat delete {vlan} vlan_name
```

To clear NAT VLAN counters, use the following command:

```
clear ip nat counters vlan {vlan_name}
```

To create or delete NAT rules, use the following commands:

```
create ip nat rule rule_name type [ source-nat | napt | destination-
napt]
```

```
delete ip nat rule rule_name
```

To configure NAT rules, use the following commands:

```
configure ip nat rule rule_name destination [[dst_ip_addr new-
destination new_dst_ip_addr {{vr} vr_name}] | none]
```

```
configure ip nat rule rule_name destination protocol [[[tcp | udp |
protocol_num] port port_num new-port new_port_num] | none]
```

```
configure ip nat rule rule_name egress {vlan} vlan_name
```

Note
All modes of NAT including destination NAPT should have the public VLAN configured as egress VLAN.

```
configure ip nat rule rule_name monitor [on | off]
```

```
configure ip nat rule rule_name name new_rule_name
```

```
configure ip nat rule rule_name source [[[src_ip_addr src_mask |
src_ipNetmask ] {{source-vr} src_vr_name} new-source new_src_ip_addr] |
none]
```

To enable or disable NAT rules, use the following commands:

```
enable ip nat rule rule_name
```

Note
The rule is programmed in hardware only when global NAT and the NAT rule are enabled.

Note
The rule configuration can be changed only when the rule is not enabled.

```
disable ip nat rule rule_name
```

To configure NAT entry aging, use the following command:

```
configure ip nat aging-time [minutes | none]
```

To display NAT information, use the following commands:

```
show ip nat vlan counters {vlan_name}
```

```
show ip nat vlan {vlan_name}
```

```
show ip nat
```

```
show ip nat rule {rule_name} statistics {no-refresh}
```

```
show ip nat rule {detail}
```

Note
NAT can be configured on 4 VLANs.

Note
1024 NAT rules are supported.

Example IP Network Address Translation (NAT) Configuration

Source NAT Rule

The following configuration is used to create a rule to translate source IP address from 10.20.30.40 to 121.144.169.196. The ingress VLAN is present in VR "VR-user-in" and the egress VLAN “internetVlan” is present in VR "VR-Internet".

```
# create ip nat rule ipOnlyRule type source-nat
# configure ip nat rule ipOnlyRule source 10.20.30.40/32 source-vr VR-user-in new-source
121.144.169.196
# configure ip nat rule ipOnlyRule egress vlan internetVlan
```

To remove the IP address configuration, you can use the following command:

```
# configure ip nat rule ipOnlyRule source none
```

Network Address Port Translation (NAPT) Rule

The following configures a base Network Address Port Translation (NAPT) rule where flows from the 10.0.0.0/8 subnet have the source IP address translated to 121.144.169.196. The L4 port number is generated internally to identify the flow:

```
# create ip nat rule naptRule type napt
# configure ip nat rule naptRule egress vlan internetVlan
# configure ip nat rule naptRule source 10.0.0.0/8 source-vr VR-user-in new-source
121.144.169.196
```

If there are flows from two hosts with IP addresses, for example, 10.1.1.214 and 10.6.9.12, two dynamic rules with names that start with SYS_NAT_RULE_XXX are created and programmed in the hardware.

If an internal server with an IP address of 10.1.1.1 listening on UDP port 80 has to be provided external access, external users can access this using an IP address of 70.1.1.254 and UDP port 8080:

Destination Network Address Port Translation Rule

```
# create ip nat rule DestNAPT type destination-napt
# configure ip nat rule DestNAPT egress vlan outVLAN
# configure ip nat rule DestNAPT destination 70.1.1.254 new-destination 10.1.1.1 vr vr-
default
# configure ip nat rule DestNAPT destination protocol udp port 8080 new-port 80
```

Limitations

NAT is not supported for the following: VXLAN tenant VLANs, MPLS service VLANs, MAC based VLANs, Netlogin VLANs, VPEX switches.

NAT is not supported with MLAG.

IP Multicast NAT

Overview

In a typical enterprise network, multicast sources use private IP addresses (RFC 1918 ranges such as 10.0.0.0/8). When multicast streams need to reach receivers in a public or external network, the source IP address must be translated to a registered public address. IP Multicast NAT performs this translation transparently at the network edge.

IP Multicast NAT is an extension of the basic source NAT (SNAT) feature. The NAT router rewrites the source IP address in the IP header of each replicated copy of a multicast packet sent to each receiver. The multicast destination group address is never modified.

Translation is keyed on the source IP address only. A single SNAT rule therefore covers all multicast groups originated by a given source host — for example, a rule matching source 10.1.1.1 applies equally to streams sent to 225.1.1.1, 225.1.1.2, and any other group from that source.

Supported Platforms

IP Multicast NAT is supported on the following platforms beginning with version 33.6.1:

- 7520
- 7720

IP Multicast NAT is not supported on stacking configurations.

Limitations

- Supported only for L3 IPv4 multicast traffic. L2 multicast and IPv6 multicast are not supported.
- Only the `source-nat` rule type is supported. The `napt` and `destination-napt` rule types do not apply to multicast traffic.

- Source IP address mask must be /32 (255.255.255.255) — one rule per source host.
- Only one VLAN can be configured as the NAT egress VLAN.
- Cannot be used concurrently with MACsec.
- Not supported on stacking platforms.
- Scalability is limited by available ACL hardware resources, as one ACL entry is installed per configured source.
- Unicast NAT and Multicast NAT are controlled by independent global enable/disable commands and can be used simultaneously or independently.

How It Works

When a multicast packet arrives on an ingress VLAN and is replicated to multiple receivers on an egress VLAN, the NAT router applies the configured SNAT rule to each outbound copy. The source IP address in each copy is replaced with the configured public address before forwarding.

For example, given:

- Source A (10.1.1.1) sending to multicast group 225.1.1.1
- Source B (10.1.1.2) sending to multicast group 225.1.1.2
- SNAT rule mapping 10.1.1.1 → 20.1.1.100 and 10.1.1.2 → 20.1.1.101

Each copy forwarded to any receiver will show source IP 20.1.1.100 (for Source A traffic) or 20.1.1.101 (for Source B traffic). The multicast group destination address (225.1.1.x) remains unchanged in all copies.

Configuration Steps

Use the following procedure to configure IP Multicast NAT:

1. Configure the ingress and egress VLANs for NAT participation:

   ```
   configure ip nat add vlan <ingress_vlan> direction ingress
   configure ip nat add vlan <egress_vlan> direction egress
   ```

2. Create a source-nat rule for each multicast source address to be translated:

   ```
   create ip nat rule <rule_name> type source-nat
   ```

3. Configure the source IP address (must use /32 mask), source VR, translated address, and egress VLAN:

   ```
   configure ip nat rule <rule_name> source <src_ip> 255.255.255.255 source-vr <vr_name>
   new-source <public_ip>
   configure ip nat rule <rule_name> egress vlan <egress_vlan>
   ```

4. Enable the rule:

   ```
   enable ip nat rule <rule_name>
   ```

5. Enable IP multicast NAT globally (and unicast NAT if also required):

   ```
   enable ip nat multicast
   enable ip nat unicast
   ```

Example Configuration

The following example configures multicast NAT for two private sources (10.1.1.1 and 10.1.1.2) that are mapped to public addresses (20.1.1.100 and 20.1.1.101) when traffic exits through `out_vlan`:

```
configure ip nat add vlan in_vlan direction ingress
configure ip nat add vlan out_vlan direction egress

create ip nat rule snat1 type source-nat
configure ip nat rule snat1 source 10.1.1.1 255.255.255.255 source-vr VR-Default new-
source 20.1.1.100
configure ip nat rule snat1 egress vlan out_vlan
enable ip nat rule snat1

create ip nat rule snat2 type source-nat
configure ip nat rule snat2 source 10.1.1.2 255.255.255.255 source-vr VR-Default new-
source 20.1.1.101
configure ip nat rule snat2 egress vlan out_vlan
enable ip nat rule snat2

enable ip nat multicast
enable ip nat unicast
```

Verification Commands

Verify global NAT and multicast NAT status:

```
show ip nat
```

Sample output showing both unicast and multicast NAT enabled:

```
Unicast NAT                          : Enabled
Multicast NAT                        : Enabled
Number of NAT VLANs                  : 2
Maximum number of NAT VLANs          : 4
Number of NAT rules                  : 2
Number of source NAT rules           : 2
Number of destination NAPT rules     : 0
Number of configured NAPT rules      : 0
Number of dynamic NAPT rules         : 0
Maximum number of NAT rules          : 1024
Aging time                           : 20 minutes
```

Display configured NAT rules and their state:

```
show ip nat rule

show ip nat rule detail
```

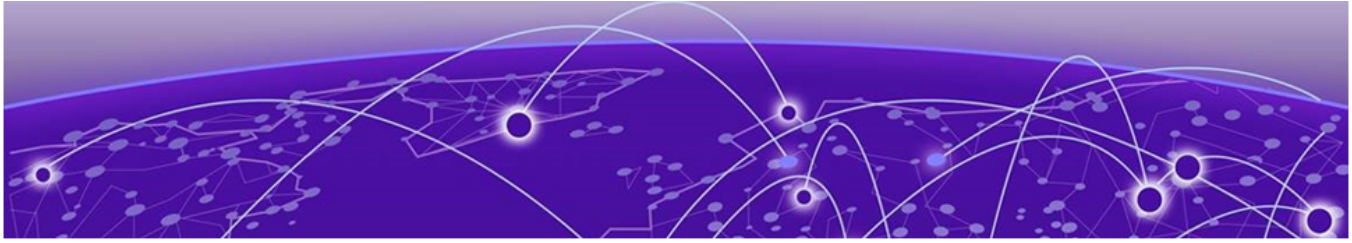
Display per-VLAN translation and discard counters:

```
show ip nat vlan counters
```

Related Commands

Command	Description
`enable ip nat multicast`	Globally enables IP multicast NAT.
`disable ip nat multicast`	Globally disables IP multicast NAT.

Command	Description
`enable ip nat [unicast]`	Globally enables IP unicast NAT (default when no keyword specified).
`create ip nat rule <name> type source-nat`	Creates a source NAT rule (the only rule type supported for multicast).
`configure ip nat rule <name> source <ip> <mask> source-vr <vr> new-source <ip>`	Configures the source address mapping for the rule. Mask must be 255.255.255.255 for multicast NAT.
`configure ip nat rule <name> egress vlan <vlan>`	Configures the egress VLAN for translated traffic.
`show ip nat`	Displays global NAT status including unicast and multicast NAT state.
`show ip nat rule {detail}`	Displays configured NAT rules and their parameters.
`show ip nat vlan counters`	Displays per-VLAN packet translation and discard counters.



IPv6 Multicast

Multicast Listener Discovery (MLD) Overview on page 945
Managing MLD on page 946

This chapter introduces IPv6 multicast, which allows a single IPv6 host to send a packet to a group of IPv6 hosts, and the features and configuration of the Multicast Listener Discovery (MLD) protocol. For more information on IPv6 multicasting, refer to the following publications:

- RFC 2710—Multicast Listener Discovery (MLD) for IPv6
- RFC 3810—Multicast Listener Discovery Version 2 (MLDv2) for IPv6

Multicast Listener Discovery (MLD) Overview

Multicast Listener Discovery (MLD) is a protocol used by an IPv6 host to register its IP multicast group membership with a router. To receive multicast traffic, a host must explicitly join a multicast group by sending an MLD report; then, the traffic is forwarded to that host. Periodically, the router queries the multicast group to see if the group is still in use. If the group is still active, a single IP host responds to the query, and group registration is maintained.

MLD is the IPv6 equivalent to *IGMP (Internet Group Management Protocol)*, and MLDv1 is equivalent to IGMPv2. The ExtremeXOS software supports both MLDv1 and MLDv2 protocol.

Note
This release does not support MVR, PVLAN, *VLAN (Virtual LAN)* Aggregation, and Multicast Troubleshooting tools for MLD/IPv6.

ExtremeXOS Resiliency Enhancement for IPv6 Static Routes

The ExtremeXOS Resiliency Enhancement feature provides a resilient way to use *ECMP (Equal Cost Multi Paths)* to load balance IPv6 traffic among multiple servers or other specialized devices. ExtremeXOS automatically manages the set of active devices using ECMP static routes configured with ping protection to monitor the health of these routes. Such servers or specialized devices do not require special software to support Bidirectional Forwarding Detection (BFD), or IP routing protocols such as *OSPF (Open Shortest Path First)*, or proprietary protocols to provide

keepalive messages. ExtremeXOS uses industry-standard and required protocols ICMPv6/Neighbor Discovery for IPv6 to accomplish the following automatically:

- Initially verify devices and activate their static routes, without waiting for inbound user traffic, and without requiring configuration of device MAC addresses.
- Detect silent device outages and inactivate corresponding static routes.
- Reactivate static routes after device recovery, or hardware replacement with a new MAC address.

ExtremeXOS currently supports similar protection and resiliency using Bidirectional Forwarding Detection (BFD) on IPv4 and IPv6 static routes. However, BFD can only be used when the local and remote device both support BFD.

Managing MLD

Enabling and Disabling MLD on a VLAN

MLD is disabled by default on all *VLAN*s. You can enable MLD using the enable `mld {vlan vlan name} {MLDv1 | MLDv2}` command.

This allows IPv6 hosts to register with IPv6 multicast groups and receive IPv6 multicast traffic.

- To disable MLD on a VLAN after it has been enabled, use the `disable mld {vlan name}` command.
- To enable MLD on a VLAN after it has been disabled, use the `enable mld {vlan vlan_name} {MLDv1 | MLDv2}` command.

MLD Snooping

Similar to *IGMP* snooping, MLD snooping is a Layer 2 function of the switch; it does not require multicast routing to be enabled. In MLD snooping, the Layer 2 switch keeps track of MLD reports and only forwards multicast traffic to the part of the local network that requires it. MLD snooping optimizes the use of network bandwidth and prevents multicast traffic from being flooded to parts of the local network that do not need it.

MLD snooping is disabled by default on all *VLAN*s in the switch.

When MLD snooping is disabled on a VLAN, all MLD and IPv6 multicast traffic floods within the VLAN.

MLD snooping expects at least one device on every VLAN to periodically generate MLD query messages.

- Enable or disable MLD snooping:

```
enable mld snooping

disable mld snooping
```

Multicast packets with a scope id less than 2 are not forwarded by the MLD snooping enabled switch. Kill entry is installed in the hardware for this traffic.

Multicast packets with a scope id of 2 and group address in the range of FF02::/111 (Addresses allocated by IANA as per RFC 3307) are always flooded to all ports of the VLAN by hardware and a copy of the packet is provided to slow path. There are no cache entries in software or hardware for these addresses.

- Multicast packets with a scope id of 2 and group address as solicited multicast address (FF02::1:FFXX:XXXX/104) are flooded to all ports of VLAN for 135 seconds (Default MLD query interval + Maximum response time), if there are no members for this group.

  Otherwise, the traffic is forwarded based on the snooping database. Multicast cache entries for these addresses are maintained only in the software and traffic is always slow path forwarded.

  Multicast addresses with a scope id of 2 and that do not qualify in the above categories will be forwarded based on the snooping database.

  Cache entries for these multicast addresses will be installed in hardware. Unregistered packets are dropped.
- In general, all multicast data traffic on a PIMv6 enabled VLAN is controlled by the PIMv6 protocol. However, multicast traffic with either the group address or source address as the link local address will not be controlled by PIMv6. Instead, it will be L2 forwarded based on the snooping database.

  For multicast packets with a scope id greater than 2 on PIMv6 enabled VLANs, cache entries are controlled by the PIMv6 protocol.

  PIMv6 provides a list of egress VLANs for which packets need to be forwarded. The snooping database is used to construct the set of ports for ingress VLANs as well as for each egress VLAN.

  On PIMv6 disabled VLANs, traffic is forwarded based on the snooping database on the ingress VLAN.

  In both cases, cache entry is installed in the hardware, and traffic is fast path forwarded.
- The MLD snooping proxy feature is enabled automatically when MLD snooping is enabled. This feature optimizes the forwarding of MLDv1 reports. The following conditions apply for each group:
  - Only the first received MLD join is forwarded upstream.
  - Only the MLD leave for last host is forwarded upstream.

    When a switch receives an MLD leave message on a port, it sends a group-specific query on that port if proxy is enabled (even if it is a non-querier). The switch removes the port from the group after the leave timeout (a configuable value from 0 - 175000ms with a default of 1000ms). If all the ports are removed from

the group, the group is deleted and the MLD leave is forwarded upstream. If MLD snooping proxy is disabled, then all the MLD reports are forwarded upstream.

Note
MLD snooping proxy does not apply to MLDv2 reports.

- MLD snooping is implemented primarily through ACLs, which are processed on the interfaces. These special purpose ACLs are called MLD snooping hardware filters. The software allows you to choose between two types of MLD snooping hardware filters: per-port filters (the default) and per-VLAN filters.

  The two types of MLD snooping hardware filters use switch hardware resources in different ways.
- The two primary hardware resources to consider when selecting the MLD snooping hardware filters are the Layer 3 multicast forwarding table, and the interface ACLs. The size of both of these hardware resources is determined by the switch model. In general, the per-port filters consume more resources from the multicast table and less resource from the ACL table. The per-VLAN filters consume less space from the multicast table and more from the ACL table.
- Using the per-port filters can fill up the multicast table and place an extra load on the CPU. To avoid this, configure the switch to use the per-VLAN filters.

Note
The impact of the per-VLAN filters on the ACL table increases with the number of VLANs configured on the switch. If you have a large number of configured VLANs, we suggest that you use the per-port filters.

MLD Snooping Filters

MLD snooping filters allow you to configure a policy file on a port to allow or deny MLD report and leave packets coming into the port.

(For details on creating policy files, see Policy Manager on page 1448.) The MLD snooping filter feature is supported by MLDv1 and MLDv2.

Note
Do not confuse MLD snooping filters with MLD snooping hardware filters explained in previous section. MLD snooping filters are software filters, and the action is applied at software level by the ExtremeXOS multicast manager.

For the policies used as MLD snooping filters, all the entries should be IPv6 address type entries, and the IPv6 address of each entry must not be in the range of FF02::/96.

Use the following template to create a snooping filter policy file that denies MLD report and leave packets for the FF03::1/128 and FF05::1/112 multicast groups:

```
#
# Add your group addresses between "Start" and "end"
# Do not touch the rest of the file!!!!
entry mldFilter {
if match any {
#------------------ Start of group addresses ------------------
nlri FF03::1/128;
```

```
nlri FF05::1/112;
#------------------- end of group addresses --------------------
} then {
deny;
}
}
entry catch_all {
if {
} then {
permit;
}
}
```

After you create a policy file, use the following command to associate the policy file and filter to a set of ports:

```
configure mld snooping vlan vlan_name ports port_list filter [policy]
```

To remove the filter, use the **none** option.

To display the MLD snooping filters, use the following command:

```
show mld snooping {vlan} name filter
```

Limiting the Number of Multicast Sessions on a Port

The default configuration places no limit on the number of multicast sessions on each *VLAN* port. To place a limit on the number of learned MLD reports, or remove the limit, use the command:

```
configure mld snooping {vlan} vlan_name ports port_list join-limit
[num_joins | no-limit]
```

Configuring MLD Snooping

- To configure the timers that control MLD operation, use the command:

  ```
  configure mld query_interval query_response_interval
  last_member_query_interval {robustness}
  ```

- Similar to *IGMP* snooping, MLD snooping is a Layer 2 function of the switch. It does not require multicast routing to be enabled. MLD snooping keeps track of MLD reports, and only forwards multicast traffic to that part of the local network that requires it. MLD snooping is disabled by default on all *VLAN*s. If MLD snooping is disabled on a VLAN, all MLD and IPv6 multicast traffic floods within the VLAN. To enable MLD snooping, use the command:

  ```
  enable mld snooping {vlan name}
  ```

Clearing MLD Group Registration

To clear a single group or all groups in a *VLAN* learned through MLD, use the command:

```
clear mld group {v6grpipaddress} {{vlan} name}
```

Configuring Static MLD

In some situations, you might want multicast traffic to be forwarded to a port where a multicast-enabled host is not available (for example, when you test multicast configurations). Static MLD emulates a host or router attached to a switch port, so that multicast traffic is forwarded to that port, and the switch sends a proxy join for all the statically configured MLD groups when an MLD query is received. You can emulate a host to forward a particular multicast group to a port; and you may emulate a router to forward all multicast groups to a port.

- To emulate a host on a port, use the command:

  `configure mld snooping {vlan} vlan_name{ ports port_list } add static group v6grpipaddress`

- To emulate a multicast router on a port, use the command:

  `configure mld snooping {vlan}vlan_name ports port_list add static router`

- To remove these entries, use the corresponding commands:

  `configure mld snooping {vlan} vlan_name {ports port_list } delete static group [all | v6grpipaddress]`

  `configure mld snooping {vlan} vlan_name ports port_list delete static router`

MLD Loopback

Prior to ExtremeXOS 15.3.2, you could configure static groups, but it was necessary to specify port(s). As of 15.3.2, the configuration of dynamic groups is now supported. The MLD Loopback feature along with the existing static group feature supports the configuration of static and/or dynamic groups with or without ports.

A *VLAN* in loopback mode may not have ports associated with it, but its operational status is up. However, it is not possible to have multicast receivers on a VLAN without having a port. Sometimes, there is a need to pull the multicast traffic from upstream to the loopback VLAN for troubleshooting. The traffic need not always be forwarded to any ports/receivers. The MLD Loopback feature allows you to configure groups on a VLAN without specifying a port, so the traffic is pulled from upstream but not forwarded to any port.

The loopback (Lpbk) port is the logical port on a VLAN in the application context. When you configure a group on a VLAN but do not specify the port, the switch forms an MLDv1 join and assumes it to be received on the Lpbk port. A dynamic group ages out after the membership timeout if there are no other receivers. Membership joins refresh the dynamic group. The static group remains until it is removed from the configuration.

The following command is used to configure static or dynamic group entry:

`configure mld snooping {vlan} vlan_name {ports portlist} add [static | dynamic] group ip_address`

Displaying MLD Information

- To display MLD configuration and operation information, use the command:
  `show mld group {{vlan} {name} | {v6grpipaddress}} {MLDv2}`
- To display the MLD static group information, use the command:
  `show igmp snooping {vlan} name static [group | router]`

MLD SSM Mapping

The MLD-SSM Mapping feature allows MLDv1 hosts to participate in SSM functionality, and eliminates the need for MLDv2. You can configure SSM map entries that specify the sources used for a group/group range for which SSM functionality is applied. You also have the option to configure domain name and DNS server to use to obtain the source list.

When a MLDV1 report is received, the configured sources are provided to PIM so that it can send source specific joins. When the host leaves, or when the membership times out, PIM is informed so that it can consider sending prunes.

Note
The sources mapped to only the LPM group are used.

Feature Implementation Information

- This feature is implemented as an extension to existing *IGMP* SSM support.
- The CLI commands for this feature are applicable in VR context.
- PIM is completely unaware of existence of this feature, so there is no change in PIM processing.
- On last hop the (S, G) cache created through MLDv1 join is similar to the (S, G) cache created as a result of RPT to SPT switchover in PIM-SM. There is no indication that the cache is created as result of MLDv1 join, or MLDv2 report.

Limitations

- Only 50 sources (static or dynamic) are allowed for each group address, or group range.

  Note
  The DNS server may send only 14 IPv6 source addresses in its response thereby limiting the number of dynamic sources supported.

- Only one DNS name is allowed for each group address/group range.

SSM Address Range

The address prefix FF3x::/32 is reserved by IANA for SSM use. All SSM addresses must have P=1, T=1 and plen=0. RFC 3306 mandates that the network prefix is zero, which results in the SSM address range to be in FF3x::/96 range. Since future documents may allow a non-zero network prefix, this feature allows the addresses in range FF3x::/32 in SSM map configuration. The default SSM range is FF3x::/96.

SSM address range is configured from the PIM context using the following command:

```
configure pim ipv6 ssm range default | policy policy_name
```

When this command is issued, PIM notifies MCMGR with the details of the SSM address range. MCMGR applies this range for the MLD SSM feature.

Handling MLD Reports

Table 61 captures the enhanced functionality.

Table 61: MLD Mapping

MLDv1 Join	MLD SSM Map Disabled	MLD SSM Map Enabled
Group in SSM range but no map entries	Dropped	Dropped
Group in SSM range with map entries	Dropped	SSM map sources included
Group not in SSM range	SSM map sources not included	SSM map sources not included

MLDv1 reduction messages in the SSM range are accepted and processed normally. Multicast manager will send out a group-specific query and refresh the receivers on receiving joins.

When an MLDv2 report is received, following group records types are ignored if they refer to SSM group range:

- MODE_IS_EXCLUDE
- CHANGE_TO_EXCLUDE_MODE

Other group records in the same report are processed.

SSM Data Forwarding

L2 forwarding decision of SSM data traffic is taken by the Multicast manager. Sources configured in the SSM map are considered only when a MLDv1 receiver is connected to a port. For the same group, there may be another port on which only MLDv2 receiver is present. For this port, SSM map sources are not considered.

L3 forwarding decision of SSM data traffic is taken by PIM. PIM provides the list of egress VLANs to be included, and the multicast manager decides the egress ports on each of these VLANs.

DNS Request/Response

When a DNS name is configured for a group in an SSM map, a DNS request is sent to resolve the name. If static sources are already configured for the group, it takes precedence and the DNS request is not sent. This feature uses the AAAA record type. For example, if the multicast group is FF30::1234:5678 and the configured domain name is ext.foo.bar, then the DNS request is sent for 8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.F.F.ext.xxx.xxx.

The DNS request is not sent every time a report is received. A DNS request is sent only in the following scenarios:

- When a DNS name is configured for a group.
- When the refresh ssm-map command is used.
- When the DNS age timer expires.

The DNS response may contain one or more addresses. These addresses are updated to the SSM mapping table against the specific group.

Note
LPM is used to match the group for a DNS response.

If a static source is configured for a group for which DNS resolved addresses are already present, the dynamic sources are flushed and the static sources are added. Static mapped addresses are preferred over DNS resolved addresses.

DNS Timers

The DNS age timer is a decrementing timer to zero and applies to a group/group range. The value of this timer is calculated by the following formula:

DNS age = (value of TTL in the DNS response) + (wait TTL)

wait TTL = 260 seconds

A DNS request is sent when receiving an MLDv1 report, but the TTL has expired. The source is not removed until the wait TTL time expires.

On receiving the DNS response, the existing DNS resolved addresses which are not in the DNS response are removed. Any new addresses in the DNS response are added and the DNS age timer is refreshed. The L2 SSM data caches are modified for the addresses removed, or added. PIM is notified to take action on L3 SSM caches.

If the DNS response is not received and the DNS age timer expires, the mapping entry is removed (if there are no static addresses). The SSM data traffic forwarding is stopped immediately when the group is removed.

DNS Server

This feature does not check or track DNS servers configured in the switch. You must correctly configure and administer the DNS server.

The following command is used to configure the DNS server:

```
configure dns-client add name-server ip_address {vr vr_name
```

The server(s) are displayed using the following command:

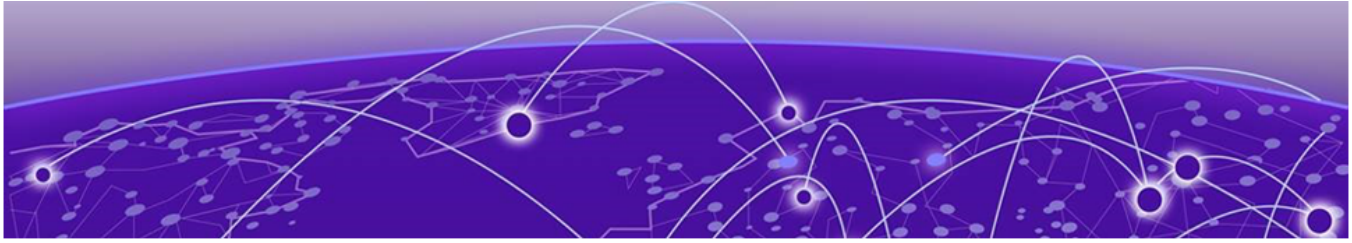
```
# show dns-client
Number of domain suffixes: 0
Number of domain servers: 1
Name Server 0: 10.120.89.75 VR-Default
```

MCMGR uses nettools library to perform DNS lookups. gethostbyname_c is used by specifying the callback function to be invoked when DNS response is received.

Configuring MLD SSM Mapping

Use the following commands to configure MLD SSM Mapping in ExtremeXOS:

- Enable or disable MLD SSM Mapping on a VR: `[enable | disable] mld ssm-map { {vr} vrname }`
- Add an MLD SSM Mapping entry on a VR: `configure mld ssm-map add v6groupnetmask [v6sourceip | src_domain_name] { {vr} vrname}`
- Delete an MLD SSM Mapping entry on a VR:`configure mld ssm-map delete v6groupnetmask [v6sourceip | src_domain_name] { {vr} vrname}`
- Delete all MLD SSM Mapping entries on a VR: `unconfigure mld ssm-map { {vr} vrname}`
- Display the status of MLD-SSM mapping feature on a VR, and display the MLD-SSM mapping entries: `show mld ssm-map { v6groupnetmask } { {vr} vrname }`
- Send out a DNS request for a particular group. On receiving the DNS response, the "DNS Age" in the SSM mapping entry is refreshed: `refresh mld ssm-map v6groupnetmask { {vr} vrname}`
- Configure the DNS server: `configure dns-client add name-server ip_address {vr vr_name}]`
- Display the DNS Servers: `show dns-client`



IPv6 Unicast Routing

IPv6 Unicast Overview on page 955
Neighbor Discovery Protocol on page 959
Managing Duplicate Address Detection on page 964
Managing IPv6 Unicast Routing on page 969
IPv6 ECMP and 32-Way ECMP on page 973
DHCPv6 Relay Remote-ID Option on page 974
DHCPv6 Relay Agent Prefix Delegation on page 975
DHCPv6 Client on page 978
Configuring DHCPv6 BOOTP Relay on page 981
Configure Route Compression on page 982
Hardware Forwarding Behavior on page 982
Routing Configuration Example on page 984
Tunnel Configuration Examples on page 985

This chapter assumes that you are already familiar with IPv6 unicast routing. If not, refer to the following publications for additional information:

- RFC 2460—Internet Protocol, Version 6 (IPv6) Specification
- RFC 4291—IP Version 6 Addressing Architecture

For IPv6 PIM support, see Multicast Routing and Switching on page 1157.

Note
For more information on interior gateway protocols, see RIPng or IPv6 Unicast Routing.

IPv6 Unicast Overview

The switch provides full Layer 3, IPv6 unicast routing. It exchanges routing information with other routers on the network using the IPv6 versions of the following protocols:

- *RIPng (Routing Information Protocol Next Generation)*
- *OSPFv3 (Open Shortest Path First version 3)*
- Intermediate System-Intermediate System (IS-IS)
- *BGP (Border Gateway Protocol)*

The switch dynamically builds and maintains a routing table and determines the best path for each of its routes.

The ExtremeXOS software can provide both IPv4 and IPv6 routing at the same time. Separate routing tables are maintained for the two protocols. Most commands that require you to specify an IP address can now accept either an IPv4 or IPv6 address and act accordingly. Additionally, many of the IP configurations, enabling, and display commands have added tokens for IPv4 and IPv6 to clarify the version for which the command applies. For simplicity, existing commands affect IPv4 by default and require you to specify IPv6, so configurations from an earlier release will still correctly configure an IPv4 network.

ACLs and routing policies also support IPv6. Use of an IPv6 address in a rule entry will automatically use IPv6.

Router Interfaces

The routing software and hardware routes IPv6 traffic between router interfaces. A router interface is either a virtual LAN (*VLAN (Virtual LAN)*) that has an IP address assigned to it, or a Layer 3 tunnel.

As you create VLANs and tunnels with IP addresses, you can also choose to route (forward traffic) between them. Both the VLAN switching and IP routing function occur within the switch.

IPv4 and IPv6 interfaces can coexist on the same VLAN, allowing both IPv4 and IPv6 networks to coexist on the same Layer 2 broadcast domain.

Note
Each IP address and mask assigned to a VLAN must represent a unique IP subnet. You cannot configure the same IP address and subnet on different VLANs within the same .

Tunnels

The ExtremeXOS software supports Layer 3 tunnels, which serve as a transition option (6in4 and 6to4 tunnels only), as networks change over from IPv4 to IPv6. The software supports these tunnels in Default-VR.

Note
IPv6 tunnels are supported only in Default-VR.

The ExtremeXOS software supports the use of IPv6-in-IPv4 tunnels (known as configured tunnels or 6in4 tunnels) and IPv6-to-IPv4 tunnels (known as 6to4 tunnels). Both types of tunnels are used to connect regions of IPv6 routing across a region of IPv4 routing. From the perspective of the router, the tunnel across the IPv4 region is one hop, even if multiple IPv4 routers are traversed during transport.

A 6in4 tunnel connects one IPv6 region to one other IPv6 region.

Multiple 6in4 tunnels can be configured on a single router to connect with multiple IPv6 regions. Dynamic and static routing can be configured across a 6in4 tunnel. Hosts in the IPv6 regions need not know anything about the configured tunnel, since packets

destined for remote regions are sent over the tunnel like any other type of routing interface.

A 6to4 tunnel connects one IPv6 region with multiple IPv6 regions. Only one 6to4 tunnel can be configured on a single router.

GRE (Generic Router Encapsulation) is a tunneling mechanism where the customer network IP packet is encapsulated by a IP+GRE header across a network infrastructure. The decapsulation is done at the remote gateway providing the original IP datagram delivered to the peers local network. The GRE tunnel is a point-to-point path between two sites, in most cases the tunnel is used to enable communication between sites using private address spaces across a service provider network or third-party network.

Specifying IPv6 Addresses

IPv6 Addresses are 128 bits (16 bytes) when compared to the 32 bit IPv4 addresses. The ExtremeXOS software accepts two standard representations for IPv6 addresses, as described in RFC 3513, section 2.2, items 1, 2, and 3.

For example, the 128 bits of the address can be represented by eight, four-digit hexadecimal numbers separated by colons:

```
2000:af13:ee10:34c5:800:9192:ba89:2311
3f11:5655:2300:304:0000:0000:7899:acde
```

Leading zeros in a four-digit group can be omitted.

There is a special use of a double colon (::) in an address. The double colon stands for one or more groups of 16 bits of zeros and can only be used once in an address. For example, the following addresses:

```
fe80:0:0:0:af34:2345:4afe:0
fe80:0:0:111:0:0:0:fe11
3c12:0:0:0:0:89:ff:3415
```

can be represented as:

```
fe80::af34:2345:4afe:0
fe80:0:0:111::fe11
3c12::89:ff:3415
```

Additionally, you can specify an address in a mixed IPv4/IPv6 mode that uses six, four-digit hexadecimal numbers for the highest-order part of the address, and uses the IPv4 dotted decimal representation for the lowest-order remaining portion.

For example:

```
0:0:0:0:0:0:192.168.1.1
0:0:0:0:0:ffff:10.0.14.254
```

These can be represented as:

```
::192.168.1.1
::ffff:10.0.14.254
```

Both global and link-local IP addresses can be configured on a *VLAN* or tunnel interface, using the following commands:

```
configure [{vlan} vlan_name | {tunnel} tunnel_name] ipaddress [ {eui64}
ipv6_address_mask | ipv6-link-local
```

```
configure tunnel tunnel_name ipaddress [ipv6-link-local | {eui64}
ipv6_address_mask ]
```

where *ipaddress* refers to the address specified in the above format.

The IPv6 address configuration can be verified using the following commands:

```
show vlan {vlan_name}
```

```
show ipconfig ipv6 {vlan vlan_name | tunnel tunnelname}
```

```
show ipconfig ipv6 {vlan vlan_name | tunnel tunnelname}
```

Scoped Addresses

IPv6 uses a category of addresses called link-local addresses that are used solely on a local subnet. Every IPv6 *VLAN* must have at least one link-local address. If a global IP address is configured on a VLAN that has no link-local address, one is assigned automatically by the switch. The link-local addresses start with the prefix fe80::/64. As a result, a switch can have the same link local address assigned to different VLANs, or different neighbors on different links can use the same link local address. Because of this, there are cases where you need to specify an address and a VLAN/tunnel to indicate which interface to use. For those cases, you can indicate the interface by using a scoped address. To scope the address, append the VLAN/tunnel name to the end of the address, separated by a percent sign (%). For example, to indicate the link local address fe80::2 on VLAN finance, use the following form:

```
fe80::2%finance
```

Scoped addresses also appear in the output of display commands.

IPv6 Addresses Used in Examples

For the purposes of documentation, we follow RFC 3849, which indicates that the prefix 2001:db8::/32 can be used as a global unicast address prefix and will not be assigned to any end party.

Neighbor Discovery Protocol

The Neighbor Discovery Protocol (NDP), as defined in RFC 2461, defines mechanisms for the following functions:

- Resolving link-layer addresses of the IPv6 nodes residing on the link.
- Locating routers residing on the attached link.
- Locating the address prefixes that are located on the attached link.
- Learning link parameters such as the link MTU, or Internet parameters such as the hop limit value that has to be used in the outgoing packets.
- Automatic configuration of the IPv6 address for an interface.
- Detecting whether the address that a node wants to use is already in use by another node, also known as Duplicate Address Detection (DAD).
- Redirecting the traffic to reach a particular destination through a better first-hop.

In IPv4, MAC address resolution is done by ARP. For IPv6, this functionality is handled by NDP. The router maintains a cache of IPv6 addresses and their corresponding MAC addresses and allows the system to respond to requests from other nodes for the MAC address of the IPv6 addresses configured on the interfaces.

Also supported is router discovery—the ability to send out router advertisements that can be used by a host to discover the router. The advertisements sent out contain the prefixes and configuration parameters that allow the end nodes to auto-configure their addresses. The switch also responds to requests from nodes for router advertisements.

The following settings can be configured on an interface to manage router advertisements:

- Settings to control the sending of router advertisements over the interface periodically and to control responding to router solicitations
- The maximum time between sending unsolicited router advertisements
- The minimum time between sending unsolicited router advertisements

You can configure the following values, that are advertised by the switch:

- Managed address configuration flag
- Other stateful configuration flag
- Link MTU
- Retransmit timer
- Current hop limit
- Default lifetime
- Reachable time

Additionally, you can configure the following values for each prefix on the prefix list associated with an interface:

- Valid lifetime of the prefix
- On-link flag

- Preferred lifetime of the prefix
- Autonomous flag

Note
ExtremeXOS software does not support host processing of neighbor router advertisements.

Managing Neighbor Discovery

Create and Delete Static Entries

- Statically configure the MAC address of IPv6 destinations on the attached links.

  ```
  configure neighbor-discovery cache {vr vr_name} add [ipv6address | scoped_link_local] mac

  configure neighbor-discovery cache {vr vr_name} delete [ipv6address | scoped_link_local]
  ```

Configure the Neighbor-Discovery Cache Size

- Configure the maximum number of entries for the neighbor-discovery cache.

  ```
  configure neighbor-discovery cache max_entries max_entries
  ```

- Configure the maximum number of pending entries for the neighbor-discovery cache.

  ```
  configure neighbor-discovery cache max_pending_entries max_pending_entries
  ```

Manage Neighbor-Discovery Cache Updates

- Configure the timeout period for dynamic entries in the neighbor-discovery cache.

  ```
  configure neighbor-discovery cache {vr vr_name} timeout timeout
  ```

- Enable the refresh of dynamic entries in the neighbor-discovery cache before the timeout period ends.

  ```
  enable neighbor-discovery {vr vr_name} refresh
  ```

- Disable the refresh of dynamic entries in the neighbor-discovery cache before the timeout period ends.

  ```
  disable neighbor-discovery {vr vr_name} refresh
  ```

Clear the Neighbor-Discovery Cache

- The neighbor-discovery entries that are learned dynamically can be cleared using the following command:

  ```
  clear neighbor-discovery cache ipv6 {ipv6address {vr vr_name} | vlan vlan_name | vr vr_name} refresh
  ```

  The above CLI command is the IPv6 version of the command to clear IPv6 neighbor discovery entries in the IPv6 neighbor table. An enhancement added in ExtremeXOS 15.7 adds a refresh option so that when “clear neighbor-discovery refresh” is executed, neighbor solicitation will be sent out for each neighbor in the

IPv6 neighbor discovery table and only active neighbors will be kept in the neighbor discovery table after the command is completed,

Return to the Neighbor-Discovery Cache Default Configuration

- Return to the neighbor-discovery cache default configuration.

  ```
  unconfigure neighbor-discovery cache {vr vr_name}
  ```

Display Neighbor-Discovery Cache Entries

- Both statically configured and dynamic neighbor-discovery entries can be viewed using the following command:

  ```
  show neighbor-discovery {cache {ipv6}} {[ipv6_addr | mac | permanent]
  {vr vr_name}} | vlan vlan_name | vr vr_name}
  ```

IPv6 Router Advertisement Options for DNS

Neighbor Discovery (ND) for IP version 6 and IPv6 stateless address autoconfiguration provide ways to configure either fixed or mobile nodes with one or more IPv6 addresses, default routers, and other parameters [RFC4861][RFC4862]. ExtremeXOS now supports two RA options that provide the DNS information needed for an IPv6 host to reach Internet services.

The Recursive DNS Server (RDNSS) option contains the addresses of recursive DNS servers, and the DNS Search List (DNSSL) option for the Domain Search List that maintains parity with the DHCPv6 options, and ensures the necessary functionality to determine the search domains.

The RDNSS option contains one or more IPv6 addresses of recursive DNS servers. All of the addresses share the same lifetime value. If you wish to have different lifetime values, you can use multiple RDNSS options.

The DNSSL option contains one or more domain names of DNS suffixes. All of the domain names share the same lifetime value. If you wish to have different lifetime values, you can use multiple DNSSL options.

The existing ND message (i.e., Router Advertisement) is used to carry this information. An IPv6 host can configure the IPv6 addresses of one or more RDNSSs through RA messages. Using the RDNSS and DNSSL options, along with the prefix information option based on the ND protocol, an IPv6 host can perform the network configuration of its IPv6 address and the DNS information simultaneously without needing DHCPv6 for the DNS configuration. The RA options for RDNSS and DNSSL can be used on any network that supports the use of ND.

For IPv6-only networks that rely only on IPv6 stateless Autoconfiguration as a deployment model, these two options allow a host to configure its DNS information. This is useful when there is no DHCPv6 infrastructure, or hosts do not have a DHCPv6 client. For networks where DHCPv6 is deployed, you might not need an RA-based DNS configuration.

You can configure the RA options for DNS using the following commands:

Default RDNSS Lifetime

```
configure {vlan} vlan_name> router-discovery {ipv6} rdnss-lifetime
[<rdnss_lifetime> | infinity | auto]
```

Add RDNS Server with Optional RDNSS Lifetime

```
configure {vlan} vlan_name router-discovery {ipv6} add rdnss ipaddress
{{rdnss-lifetime} [rdnss_lifetime | infinity]}
```

Default DNSSL Lifetime

```
configure {vlan} vlan_name router-discovery {ipv6} dnssl-lifetime
[dnssl_lifetime | infinity | auto]
```

Add a DNS suffix to DNSSL

```
configure {vlan} vlan_name router-discovery {ipv6} add dnssl dns_suffix
{{dnssl-lifetime} [dnssl_lifetime | infinity]}
```

IPv6 Router Advertisement Filtering

Newly connected IPv6 hosts can automatically discover configuration details by sending link-local router solicitation messages. Once received, an IPv6 router can send IPv6 router advertisements (RAs) back to the host, that include network configuration parameters. Since IPv6 router advertisement packets are used to configure, among other things, the host's gateway address, unauthorized users employ various methods to spoof IPv6 router advertisements to redirect, or deny service.

The IPv6 Router Advertisement Filtering feature exposes the existing "icmp-type" match criteria to allow IPv6 packets to match RAs and other IPv6 packets that use the ICMPv6 protocol. This functionality provides the ability to flexibly detect certain conditions and take appropriate actions based on network design and expectations.

Limitations

This feature has the following limitations:

- Only ingress *ACL (Access Control List)*s support the "icmp-type" match criteria for IPv6 packets. This match criteria cannot be used with egress ACLs.
- The IPv6 extension header parsing varies per platform - see "Platforms Supported" section for more detail
- The IPv6 source-address and destination-address, and the ethernet-source-address and ethernet-destination-address fields cannot be matched in the same rule without enabling "double-wide" mode. Double wide mode is not available on all of the supported platforms and causes a 50% reduction of ACL hardware resources.

Supported Platforms

All ExtremeSwitching platforms are supported. However, no IPv6 extension headers can be parsed while still matching the supplied *ICMP (Internet Control Message Protocol)* type:

The exact field compatibility with this match criteria depends on the platform, but all platforms are able to match the port and protocol (ICMPv6) in single wide mode. Using double wide mode provides access to a 128-bit source address, or source MAC address, for example. All of the above platforms support double wide mode at the expense of reducing ACL scale by 50%. The XGS3 platforms do not support double wide mode at all.

On platforms that support double wide mode, if the layer-2 device is unable to identify whether the packet is an ICMPv6 Router Advertisement message, and the IPv6 Source Address of the packet is a link-local address or is unspecified, the packet is blocked.

You can also use the new “icmp-type” to match other protocol cases such as MLD and MLDv2.

CLI Commands

The existing ACL match criteria `icmp-type` *`type`* is exposed in dynamic ACLs and static ACL policies on the target platforms. This same match criteria is already supported for rules that specify IPv4 criteria on the target platforms.

Example Policy

Here is a policy to detect and log (rate limit the same by 1 log per minute) a “simple” RA attack, only allow TCP/UDP/ICMP/xyz protocol traffic that can be parsed (i.e., has up to 2 extension headers and, if fragmented, the L4 NH is contained in the first fragment), and deny everything else including “complex” RA attacks:

```
entry disallow_and_log_RA_attacks
{
if
{protocol icmpv6;icmp-type 134;}
then
{ deny; mirror-cpu; log rate-limit 1 in 60; count RA_attack;}}
entry allow_tcp
{
if {protocol tcp; first-fragments;}
then {permit;}}
entry allow_udp
{
if {protocol udp; first-fragments;}
then {permit;}}
entry allow_icmp
{
if {protocol icmpv6; first-fragments;}
then {permit;}}
entry allow_xyz…entry denyall
{
if{first-fragments; }
then
{
deny;}}
```

The above policy works for newer chipsets, but leaves (at least) the following exposure for older chipsets: a malicious user could send an RA with a single extension header and it would be allowed to pass due to rule “allow_icmp” (newer chipsets would block this packet through the first rule). To mitigate this exposure on older chipsets, you

could call out each “icmp-type” that is supported (ND, MLD, etc.), and then drop any ICMPv6 with an extension header.

For more information, please refer to http://tools.ietf.org/html/draft-ietf-v6ops-ra-guard-implementation-04.

Managing Duplicate Address Detection

The Duplicate Address Detection (DAD) feature checks networks attached to a switch to see if IP addresses configured on the switch are already in use on an attached network. The following sections provide additional information on the DAD feature:

- DAD Overview on page 964
- Configure DAD
- Display DAD Configuration and Statistics on page 965
- Clear the DAD Counters

When you configure an active interface with an IPv6 address, the interface must send out an advertisement containing its address.

All other interfaces on the subnet have the opportunity to respond to the newly configured interface, and inform it that the address is a duplicate. Only after this process occurs, can the interface use the newly configured address. If the interface receives a message that the newly configured address is a duplicate, it cannot use the address.

Until the Duplicate Address Detection (DAD) process completes, the new address is considered tentative, and will be shown as such in any display output. If the address is a duplicate, it will also be labeled as such, and must be reconfigured. On an active interface, the DAD process should occur so quickly that you would not see the address labeled as tentative. However, if you are configuring an interface before enabling it, and you display the configuration, you will see that the address is currently tentative. As soon as you enable the interface, the address should be ready to use, or labeled as duplicate and must be reconfigured.

See RFC 2462, IPv6 Stateless Address Autoconfiguration, for more details.

DAD Overview

When administratively enabled on a VR, Duplicate Address Detection (DAD) runs when an IPv6 address is added to a *VLAN*, the VLAN transitions to an operational state, or when the first valid link-local address is added to the VLAN.

DAD works by sending a neighbor solicitation for each IPv6 address configured on a VLAN. If another device replies to the neighbor solicitations by sending neighbor advertisements for any of these IPv6 addresses, the referenced IPv6 addresses are marked duplicate and are not used. If no responses are received, the IPv6 addresses become active and available for subsequent use. Once an IPv6 address becomes active, DAD does not run again for that address unless that address is deleted and re-added,

the VLAN goes down and comes back up, or the first valid link-local address is added to the VLAN.

IPv6 functionality requires at least one valid link-local address. If the last valid link-local address is marked duplicate, all non-duplicate IPv6 addresses become tentative.

Configure DAD

- Enable or disable the DAD feature and configure feature operation.

  ```
  configure ipv6 dad [off | on | {on} attempts max_solicitations] {{vr} vr_name | vr all}
  ```

  Note
  For IPv6 interfaces, the DAD feature is automatically enabled on all platforms that support the Edge (or higher) or Base (or higher) licenses.

Display DAD Configuration and Statistics

- Display DAD configuration and statistics information.

  ```
  show ipv6 dad {[{{vr} vr_name {ip_address} | vr all | [ { vlan } vlan_name | vlan vlan_list ] {tentative | valid | duplicate} | {{vr} vr_name} ipaddress] | {tunnel} tunnel_name}
  ```

Clear the DAD Counters

- Clear the DAD feature statistics counters.

  ```
  clear ipv6 dad {{vr} vr_name {ipaddress} | vr all | {vlan} vlan_name} {counters}
  ```

Populating the Routing Table

The switch maintains an IP routing table for both network routes and host routes. The table is populated from the following sources:

- Dynamically, by way of routing protocol packets or by *ICMP* redirects exchanged with other routers
- Statically, by way of routes entered by the administrator:
  - Default routes, configured by the administrator
  - Locally, by way of interface addresses assigned to the system
  - By other static routes, as configured by the administrator

Once routes are populated using the above method, IPv6 forwarding needs to be enabled on the *VLAN* using the following command:

```
enable ipforwarding ipv6 {vlan vlan_name | tunnel tunnel_name | vr
vr_name}
```

Note
If you define a default route and subsequently delete the VLAN on the subnet associated with the default route, the invalid default route entry remains. You must manually delete the configured default route.

Dynamic Routes

Dynamic routes are typically learned by enabling the *RIP (Routing Information Protocol)*, *OSPF (Open Shortest Path First)*, IS-IS or *BGP* protocols, and are also learned from *ICMP* redirects exchanged with other routers. These routes are called dynamic routes because they are not a permanent part of the configuration. The routes are learned when the router starts up and are dynamically updated as the network changes. Older dynamic routes are aged out of the tables when an update for the network is not received for a period of time, as determined by the routing protocol.

Once a routing protocol is configured, dynamic routes require no configuration and are automatically updated as the network changes.

Static Routes

Static routes are routes that are manually entered into the routing tables and are not advertised through the routing protocols. Static routes can be used to reach networks that are not advertised by routing protocols and do not have dynamic route entries in the routing tables. Static routes can also be used for security reasons, to create routes that are not advertised by the router.

Static routes are configured in the ExtremeXOS software, remain part of the configuration when the switch is rebooted, and are immediately available when the switch completes startup. Static routes are never aged out of the routing table, however, the Bidirectional Forwarding Detection (BFD) feature, can be used to bring down static routes when the host link fails.

Without BFD, static routes always remain operationally active because there is no dynamic routing protocol to report network changes. This can lead to a black hole situation, where data is lost for an indefinite duration. Because upper layer protocols are unaware that a static link is not working, they cannot switch to alternate routes and continue to use system resources until the appropriate timers expire.

With BFD, a static route is marked operationally inactive if the BFD session goes down. Upper layer protocols can detect that the static route is down and take the appropriate action.

A default route is a type of static route that identifies the default router interface to which all packets are routed when the routing table does not contain a route to the packet destination. A default route is also called a default gateway.

ExtremeXOS Resiliency Enhancement for IPv6 Static Routes

The ExtremeXOS Resiliency Enhancement feature provides a resilient way to use *ECMP (Equal Cost Multi Paths)* to load balance IPv6 traffic among multiple servers

or other specialized devices. ExtremeXOS automatically manages the set of active devices using ECMP static routes configured with ping protection to monitor the health of these routes. Such servers or specialized devices do not require special software to support Bidirectional Forwarding Detection (BFD), or IP routing protocols such as *OSPF*, or proprietary protocols to provide keepalive messages. ExtremeXOS uses industry-standard and required protocols ICMPv6/Neighbor Discovery for IPv6 to accomplish the following automatically:

- Initially verify devices and activate their static routes, without waiting for inbound user traffic, and without requiring configuration of device MAC addresses.
- Detect silent device outages and inactivate corresponding static routes.
- Reactivate static routes after device recovery, or hardware replacement with a new MAC address.

ExtremeXOS currently supports similar protection and resiliency using Bidirectional Forwarding Detection (BFD) on IPv4 and IPv6 static routes. However, BFD can only be used when the local and remote device both support BFD.

Multiple Routes

When there are multiple, conflicting choices of a route to a particular destination, the router picks the route with the longest matching network mask. If these are still equal, the router picks the route using the following default criteria (in the order specified):

- Directly attached network interfaces
- Static routes
- *ICMP* redirects
- Dynamic routes
- Directly attached network interfaces that are not active.

Note
If you define multiple default routes, the route that has the lowest metric is used. If multiple default routes have the same lowest metric, the system picks one of the routes.

The criteria for choosing from multiple routes with the longest matching network mask is set by choosing the relative route priorities.

Relative Route Priorities

Table 62 lists the relative priorities assigned to routes depending on the learned source of the route.

Note
Although these priorities can be changed, do not attempt any manipulation unless you are expertly familiar with the possible consequences.

Table 62: Relative Route Priorities

Route Origin	Priority
Direct	10
BlackHole	50
Static	1100
HostMobility	1150
ICMP	1200
Auto-peering	1699
OSPFv3Intra	2200
OSPFv3Inter	2300
IS-IS	2350
IS-IS L1	2360
IS-IS L2	2370
RIPng	2400
OSPFv3 ASExt	3100
OSPFv3 Extern1	3200
OSPFv3 Extern2	3300
IS-IS L1Ext	3400
IS-IS L2Ext	3500
EBGP	1700
IBGP	1900

To change the relative route priority, use the following command:

```
configure iproute ipv6 priority [auto-peering | ripng | blackhole |
icmp | host-mobility | static | ospfv3-intra | ospfv3-inter | ospfv3-as-
external | ospfv3-extern1 | ospfv3-extern2 | isis |isis-level-1 | isis-
level-2 | isis-level-1-external | isis-level-2-external | ebgp | ibgp]
priority {vr vr_name}
```

Route priority configurable value is between 11-65,534.

Unique Local Address (ULA) for IPv6

RFC 4193 defines a globally unique address that is used for local communications. For instance, they are routable within a specified site or between sites. These ULAs are similar to private addresses in IPv4 (RFC 1918).

Of the 128 bits available in IPv6 address, the last 64-bits are used as the interface ID.

The remaining 64 bits are used as follows:

First 8 bits are used to define the known prefix FC00::/7 or FD00::/7

40 bits - global ID - Generated using a random algorithm specified in the RFC 4193. ExtremeXOS expects the operator to specify the 40-bit Global ID as ULA address management becomes easier, especially a mult-vendor environment.

16 bits - Used to create subnets within the site.

ULA prefixes should not be accepted by the border routers.

Additionally, ULA prefixes should not be advertised by the border routers. In ExtremeXOS, for *BGP* and *OSPFv3*, you must specify the policies to filter ULA prefixes.

There are no new CLI commands introduced to configure ULAs in ExtremeXOS. You can use the following existing command to configure a ULA:

```
configure {vlan} vlan_name ipaddress [ipaddress {ipNetmask} |
ipv6-link-local | {eui64} ipv6_address_mask]
```

Here is an example of the command:

```
configure vlan v1 ipaddress fd21:0941:2c55::/48
```

The scope of ULA is global by default in RFC 4193.

All applications treat these addresses in a similar manner as any other type of global IPv6 unicast addresses.

Managing IPv6 Unicast Routing

You should be familiar with IP Route Sharing for both IPv4 and IPv6. For information about IP route sharing, see Configuring IP Route Sharing on page 909.

For SummitStack and ExtremeSwitching series switches that support Layer 3 routing, the ExtremeXOS software supports route sharing across up to 2, 4, 8, 16, or 32 next-hop gateways.

Enabling Route Sharing for IPv6

To enable route sharing for IPv6, use the following command:

```
enable iproute ipv6 sharing
```

Configuring Basic IP Unicast Routing

To configure basic IP unicast routing, do the following:

1. Create and configure two or more *VLAN*s.
2. Assign each VLAN that will be using routing an IP address using the following command:

   `configure {vlan} vlan_name ipaddress [ipaddress {ipNetmask} | ipv6-link-local | {eui64} ipv6_address_mask]`

   Ensure that each VLAN has a unique IP address.
3. Configure a static route using the following command:

   `configure iproute add ipv6Netmask [ipv6Gateway | ipv6ScopedGateway] {metric} {vr vr_name} {multicast | multicast-only | unicast | unicast-only}`

   or

   Configure a default route using the following command:

   `configure iproute add default [{gateway {metric} {vr vr_name} {unicast-only | multicast-only}} | {lsp lsp_name {metric}}]`

   Default routes are used when the router has no other dynamic or static route to the requested destination.
4. Turn on IP routing for one or all VLANs using the following command:

   `enable ipforwarding ipv6 {vlan vlan_name | tunnel tunnel_name | vr vr_name}`
5. Configure the routing protocol, if required. For a simple network using *RIPng*, default configuration might be acceptable.
6. Turn on RIPng or *OSPFv3* using one of the following commands:

   `enable ripng`

   `enable ospfv3`

   Note

   BGP does not use ECMP by default, so if you require that functionality you must explicitly issue the `configure bgp maximum-paths max-paths` command with a value greater than 1.

Managing Router Discovery

Enable and Disable Router Discovery

- Enable or disable router discovery on a *VLAN*.

  `enable router-discovery {ipv6} vlan vlan_name`

  `disable router-discovery {ipv6} vlan vlan_name`

Add and Delete Prefixes for Router Discovery

- Add or delete prefixes for advertisement by router discovery.

  ```
  configure vlan vlan_name router-discovery {ipv6} add prefix prefix

  configure vlan vlan_name router-discovery {ipv6} delete prefix [prefix | all]
  ```

Configure Router Discovery Settings

- Configure the router discovery settings using the following commands:
  - `configure vlan vlan_name router-discovery {ipv6} default-lifetime defaultlifetime`
  - `configure vlan vlan_name router-discovery {ipv6} link-mtu linkmtu`
  - `configure vlan vlan_name router-discovery {ipv6} managed-config-flag on_off`
  - `configure vlan vlan_name router-discovery {ipv6} max-interval maxinterval`
  - `configure vlan vlan_name router-discovery {ipv6} min-interval mininterval`
  - `configure vlan vlan_name router-discovery {ipv6} other-config-flag on_off`
  - `configure vlan vlan_name router-discovery {ipv6} reachable-time reachabletime`
  - `configure vlan vlan_name router-discovery {ipv6} retransmit-time retransmittime`
  - `configure vlan vlan_name router-discovery {ipv6} set prefix prefix [autonomous-flag auto_on_off | onlink-flag onlink_on_off | preferred-lifetime preflife |valid-lifetime validlife]`
- Reset all router discovery settings to their default values using the command: `unconfigure vlan vlan_name router-discovery {ipv6}`
- To reset an individual router discovery setting to its default value, enter one of the following commands:
  - `unconfigure vlan vlan_name router-discovery {ipv6} default-lifetime`
  - `unconfigure vlan vlan_name router-discovery {ipv6} hop-limit`
  - `unconfigure vlan vlan_name router-discovery {ipv6} link-mtu`
  - `unconfigure vlan vlan_name router-discovery {ipv6} managed-config-flag`
  - `unconfigure vlan vlan_name router-discovery {ipv6} max-interval`
  - `unconfigure vlan vlan_name router-discovery {ipv6} min-interval`
  - `unconfigure vlan vlan_name router-discovery {ipv6} other-config-flag`
  - `unconfigure vlan vlan_name router-discovery {ipv6} reachable-time`
  - `unconfigure vlan vlan_name router-discovery {ipv6} retransmit-time`

Display Router Discovery Configuration Settings

To display router discovery settings, use the command: `show router-discovery {ipv6} {vlan vlan_name}`

Managing Tunnels

IPv6-in-IPv4 and IPv6-to-IPv4 tunnels are introduced in Tunnels on page 956.

Create an IPv6-in-IPv4 Tunnel

- Create an IPv6-in-IPv4 tunnel.

  `create tunnel tunnel_name ipv6-in-ipv4 destination destination-address source source-address`

  The *source-address* refers to an existing address in the switch, and the *destination-address* is a remote destination accessible from the switch. A maximum of 255 IPv6-in-IPv4 tunnels can be configured.

Create an IPv6-in-IPv4 Tunnel

- Create an IPv6-in-IPv4 tunnel.

  `create tunnel tunnel_name ipv6-in-ipv4 destination destination-address source source-address`

  The *source-address* refers to an existing address in the switch, and the *destination-address* is a remote destination accessible from the switch. A maximum of 255 IPv6-in-IPv4 tunnels can be configured.

Delete a Tunnel

To delete a tunnel, use the command: `delete tunnel tunnel_name`

Configure an IPv6 Address for a Tunnel

To configure or unconfigure IPv6 addresses for the tunnels, use the commands:

`configure tunnel tunnel_name ipaddress [ipv6-link-local | {eui64} ipv6_address_mask ]`

`unconfigure tunnel tunnel_name ipaddress ipv6_address_mask`

Display Tunnel Information

- Display tunnel information.

  `show [{tunnel} {tunnel_name}]`

Verifying the IP Unicast Routing Configuration

- To display the currently configured routes, which includes how each route was learned, use the command: `show iproute ipv6`

  Additional verification commands include:

  - `show neighbor-discovery cache ipv6`—Displays the neighbor discovery cache of the system.
  - `show ipconfig ipv6`—Displays configuration information for one or more VLANs.
  - `show ipstats ipv6`—Displays the IPv6 statistics for the switch for the specified VLANs.

Managing IPv6 Routes and Hosts in External Tables

When external LPM tables are supported and configured on a switch, the configuration setting applies only to the external LPM tables. You can configure the external LPM to contain both IPv4 and, or, IPv6 routes. Internal LPM tables only store IPv4 routes.

The configure forwarding external-tables command using the ipv6 and ipv4-and-ipv6 variables supports larger IPv6 route and host scaling in external LPM tables.

When an external LPM table is configured for l3-only ipv6, no IPv6 routes or IPv6 hosts are configured in any of the internal hardware tables. This provides the highest IPv6 scale, and avoids contention with IP Multicast in the L3 Hash hardware table.

IPv6 hardware and slowpath forwarding are supported on user-created Virtual Routers, and IPv6 tunnels are only supported on *VR-Default*.

The size of the internal LPM tables, and the size of the L3 Hash and Next Hop tables are fixed. For specific information on hardware capacity, refer to the following table, the following table, and the following table in IPv4 Unicast Routing.

IPv6 ECMP and 32-Way ECMP

This feature adds IPv6 Equal-Cost Multi-Path (ECMP) support on several new platforms. Additionally, it adds support for 16-way and 32-way ECMP (for both IPv4 and IPv6), using static routes, and up to 8-way ECMP for IPv4 and IPv6 using routing protocols .

Sharing ECMP gateway sets now applies to IPv6 as well as IPv4. Sharing ECMP gateway sets for IPv6 means the entire IPv6 Longest-Prefix Match (LPM) hardware capacity can use ECMP, across up to 32 gateways.

Feature Description

This feature adds IPv6 ECMP support on all platforms.

Additionally, ExtremeXOS supports 16-way and 32-way ECMP for both IPv4 and IPv6 using static routes, and up to 8-way ECMP for IPv4 and IPv6 using routing protocols. Now, the maximum number of gateways in each IPv4 or IPv6 gateway set in the ECMP hardware table can be configured as 16 or 32.

DHCPv6 Relay Remote-ID Option

Available in ExtremeXOS 15.5, the remote-id option is added to the relay forward IPv6 message, if the specific *VLAN* has only the link local IPv6 address. This remote-id is added to all the request packets received from the client on that VLAN, if it is configured or not on the VLAN. The remoteID option added to the packet after the relay message header has the following format, as defined in RFC 4649:

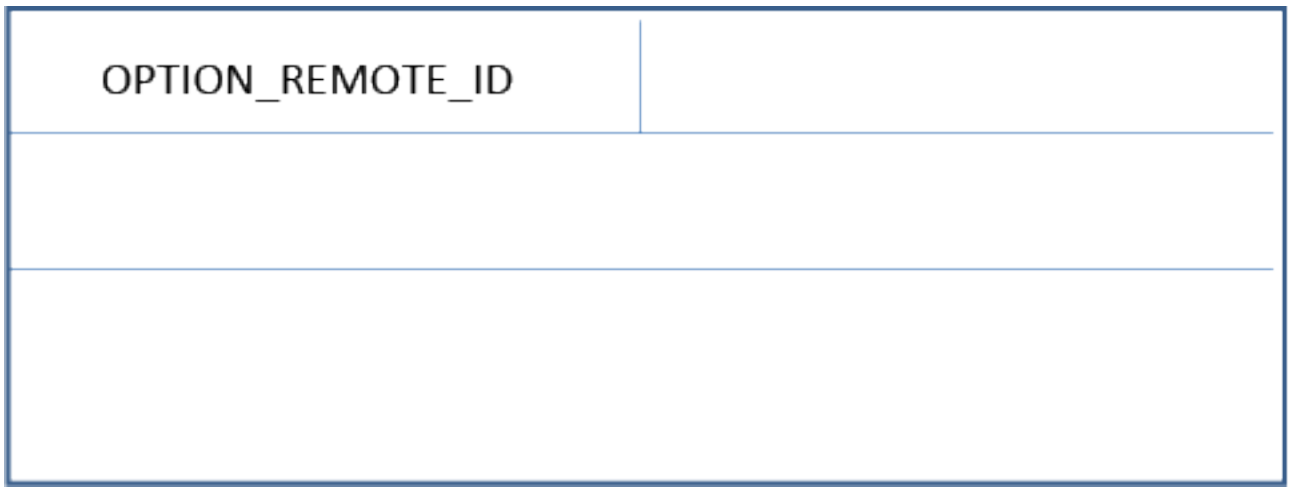


Figure 104: DHCPv6 Relay Remote-ID Option

If the remote-id is configured through the CLI on a specified VLAN, it can have either a user defined value, or the system-name. If the user does not configure the value or marks “none” as the remote-id, the switch MAC address will be used as the remote-id. The value of the remote-id present in the packet will have the following format:`vlan_port_<remoteId`, where *`remoteId`* will be either the configured string, or the system name or the switch MAC address. The Enterprise number field will be populated with the Extreme Networks enterprise number, “1916”.

The remote-id is added to the packets on a specified VLAN, only if:

- IPv6 bootprelay is enabled on the VLAN.
- rhe vlan has only the link local address.

Once a relay agent receives one of the following messages from the client or relay agent, the relay agent will verify if the specific interface has only the link local address:

- DHCPV6_SOLICIT
- DHCPV6_REQUEST
- DHCPV6_CONFIRM
- DHCPV6_RENEW
- DHCPV6_REBIND
- DHCPV6_RELEASE
- DHCPV6_DECLINE
- DHCPV6_INFORMATION_REQUEST

- DHCPV6_RELAY_FORW
- DHCPV6_LEASEQUERY

If so, it verifies if the specific VLAN has a remote-id configured. If a remote-id is present, it is added to the packet along with the VLAN and port, in the format mentioned above. If it is not configured, the default (switch MAC address) will be added as the remote id, along with the VLAN and port.

On the reverse path, if it receives any of the following messages from the server or any other relay agent, and if the interface has only a link local address, it will verify for the presence of remote-id in the packet:

- DHCPV6_ADVERTISE
- DHCPV6_REPLY
- DHCPV6_RECONFIGURE
- DHCPV6_RELAY_FORW
- DHCPV6_LEASEQUERY_REPLY

If it is not present, the agent will check for the interface-id. If one of these is present, it will verify if the value matches the local configuration. If it matches, after removing the remote-id and/or interface-id, the packet will be forwarded to the respective client. If the remote-id is present in the packet, and it does not match the configured value, the packet will be dropped. If none of them are present, the packet will be forwarded to the client, based on the IPv6 address of the VLAN.

DHCPv6 Relay Agent Prefix Delegation

DHCPv6 Prefix Delegation options provide a mechanism for automated delegation of IPv6 prefixes using *DHCP (Dynamic Host Configuration Protocol)*. This mechanism is intended for delegating a long-lived prefix from a delegating router to a requesting router, across an administrative boundary, where the delegating router does not have/require knowledge about the topology of the networks to which the requesting router is attached, and the delegating router does not require other information aside from the identity of the requesting router to choose a prefix for delegation.

For example, these options would be used by a service provider to assign a prefix to a Customer Premise Equipment (CPE) device acting as a router between the subscriber's internal network and the service provider's core network.

This prefix delegation mechanism is appropriate for use by an ISP to delegate a prefix to a subscriber, where the delegated prefix would possibly be sub-netted and assigned to the links within the subscriber's network.

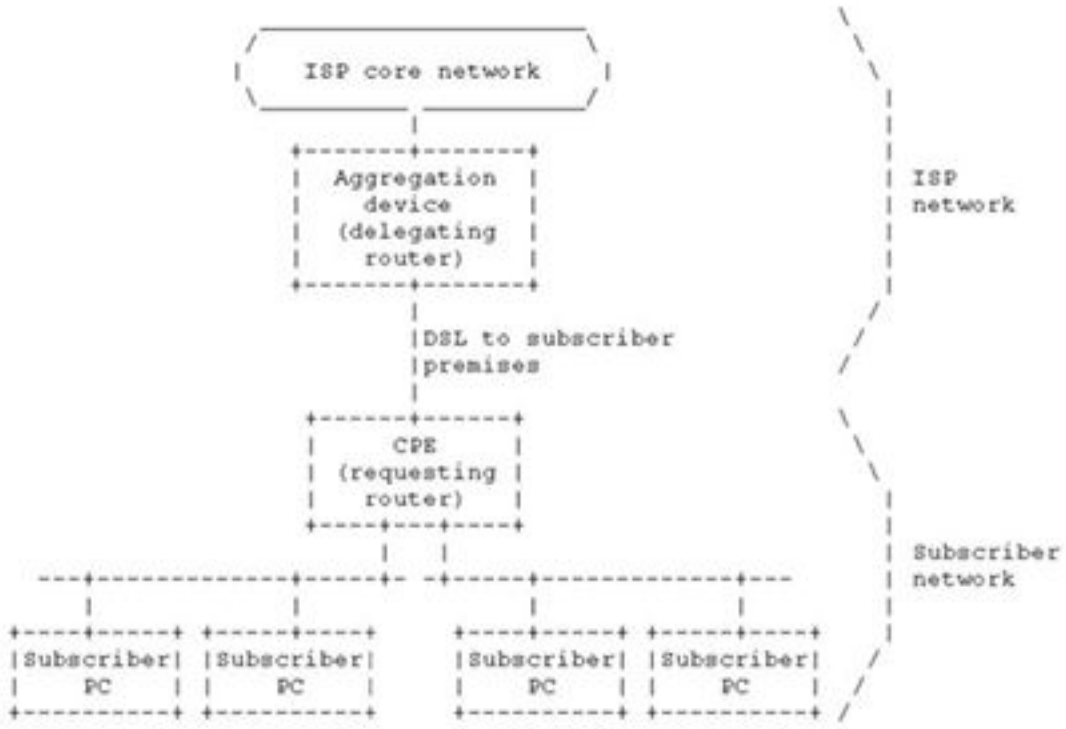


Figure 105: Prefix Delgation Example

The prefix delegation process begins when the requesting router requests configuration information through DHCP. When the delegating router receives the request, it selects and returns an available prefix or prefixes for delegation to the requesting router.

The requesting router now acts as a DHCP server for the subscriber network. It subnets the delegated prefix and assigns the longer prefixes to links in the subscriber's network. In a typical scenario based on the network shown in Figure 1, the requesting router subnets a single delegated /48 prefix into /64 prefixes and assigns one /64 prefix to each of the links in the subscriber network.

Prefix delegation with DHCP is independent of address assignment with DHCP. A requesting router can use DHCP for just prefix delegation or for prefix delegation along with address assignment and other configuration information.

Relay Agent Behavior in Prefix Delegation

A relay agent forwards *DHCP* messages containing Prefix Delegation options in the same way as other DHCP messages.

RFC 3633 says “If a delegating router communicates with a requesting router through a relay agent (sitting in between the Aggregation device (PE) and the CPE), the delegating router may need a protocol or other out-of-band communication to add routing information for delegated prefixes into the provider edge router.”

The requesting router acts as a DHCPv6 server behind the CPE and assigns IPv6 subnets to the customer devices. Since there is no mechanism to notify the delegating router about the IPv6 subnets assigned by the requesting router, the core routers have no clue about these IPv6 subnets and therefore have no dynamic route added into the routing tables in the core. RFC 3633 leaves it to the implementations to device a mechanism to solve this problem.

When the ExtremeXOS DHCPv6 relay agent relays the DHCP messages with IA_PD options and detects that a successful prefix delegation operation has been completed and a prefix or a set of prefixes have been delegated, it installs one route for each of prefixes that has been delegated. This ensures reachability between the customer devices and the service provider network.

When ExtremeXOS DHCPv6 relay agent detects a successful DHCP prefix delegation transaction (Solicit-Advertise-Request-Reply), it looks into the details of the prefix(es) that are being delegated. Each DHCP message with IA_PD option can have multiple IAPREFIX options in it. Each IAPREFIX option represents a prefix that is being delegated.

ExtremeXOS DHCPv6 relay agent installs one route for each of the delegated prefixes. The route is installed on the *VLAN* on which the requesting router (CPE) is reachable and with the requesting router (CPE) as the gateway for the route, if the requesting router is directly connected to our ExtremeXOS DHCPv6 Relay agent. If the DHCPv6 messages from the requesting router have been received via another DHCPv6 Relay agent, the route is installed with the next hop DHCPv6 Relay agent as the gateway. If the DHCPv6 packets from the next hop DHCPv6 Relay agent have been IPv6 forwarded via other routers in between (which did not do DHCPv6 Relay, but just IPv6 forwarding), the next hop router to reach the next hop DHCPv6 Relay agent is used as the gateway for the route. ExtremeXOS DHCPv6 Relay uses Route Manager (rtMgr) client library APIs for adding and deleting these routes.

To ensure that the routes are retained across reboots (or restart process netTools), ExtremeXOS DHCPv6 relay agent stores these routes along with their validity times in an internal file. After a reboot, ExtremeXOS DHCPv6 relay agent reads this file and installs the routes once again. This file will be a flat file with the following format: `vlanName ipv6Prefix ipv6GatewayAddr startTime validTime`.

Only those IPv6 prefixes which are still valid after the reboot are added again. The expired IPv6 prefixes are discarded. As the file format denotes, this file contains the details about the VLAN, the actual delegated IPv6 prefix, the IPv6 Gateway Address, the start time when it got delegated and the time for which the delegated prefix is deemed to be valid (in seconds). This file is internal to ExtremeXOS DHCPv6 relay agent and cannot/should not be edited by anyone else.

ExtremeXOS DHCPv6 relay agent checkpoints the prefix delegations that have been detected.

Basic Snooping Configuration

```
create vlan v1

enable bootprelay ipv6
```

```
conf bootprelay ipv6 prefix-del snooping on vlan v1

con bootprelay ipv6 prefix-delegation snooping add 5001:db8:3553:bf00::/56
fe80::a440:cfd5:c05b:d324 vlan v1 valid-time 300

* (Engineering) Slot-1 Sharmila_DUT1_Stack-->.13 # sh bootprelay ipv6 prefix-delegation
snooping

Delegated Prefix                                Interface
        Gateway                                         Valid For
-------------------------------------------------------------------------

5001:db8:3553:bf00::/56                         v1
       fe80::a440:cfd5:c05b:d324                              300 secs

* (Engineering) Slot-1 Sharmila_DUT1_Stack-->.14 # show iproute ipv6
Ori Destination                                 Mtr  Flags         Duration
    Gateway                                     Interface
 bo 5001:db8:3553:bf00::/56                     1    -G-D---um---- 0d:0h:0m:32s
    fe80::a440:cfd5:c05b:d324                   [VR-Default]

Origin(Ori):(b) BlackHole, (be) EBGP, (bg) BGP, (bi) IBGP, (bo) BOOTP,
            (ct) CBT, (d) Direct, (df) DownIF, (dv) DVMRP, (e1) ISISL1Ext,
            (e2) ISISL2Ext, (h) Hardcoded, (i) ICMP, (i1) ISISL1 (i2) ISISL2,
            (is) ISIS, (mb) MBGP, (mbe) MBGPExt, (mbi) MBGPInter, (ma) MPLSIntra,
            (mr) MPLSInter, (mo) MOSPF (o) OSPFv3, (o1) OSPFv3Ext1, (o2) OSPFv3Ext2,
            (oa) OSPFv3Intra, (oe) OSPFv3AsExt, (or) OSPFv3Inter, (pd) PIM-DM, (ps) PIM-
SM,
            (r) RIPng, (ra) RtAdvrt, (s) Static, (sv) SLB_VIP, (un) UnKnown,
            (*) Preferred unicast route (@) Preferred multicast route,
            (#) Preferred unicast and multicast route.

Flags: (b) BFD protection requested, (B) BlackHole, (c) Compressed Route,
       (D) Dynamic, (f) Provided to FIB, (G) Gateway, (H) Host Route,
       (l) Calculated LDP LSP, (L) Matching LDP LSP, (m) Multicast,
       (p) BFD protection active, (P) LPM-routing, (R) Modified, (s) Static LSP,
       (S) Static, (t) Calculated RSVP-TE LSP, (T) Matching RSVP-TE LSP,
       (u) Unicast, (U) Up, (3) L3VPN Route.

Mask distribution:
     1 routes at length  56

Route Origin distribution:
     1 routes from Bootp

Total number of routes = 1
```

DHCPv6 Client

Note
With ExtremeXOS 30.3, ISC DHCPv6 client version is upgraded to dhcp-4.4.1.

ExtremeXOS supports Dynamic Host Configuration Protocol version 6 (DHCPv6) clients, receiving its IPv6 address and corresponding parameters for any *VLAN* interface from an external DHCPv6 server. For the ExtremeXOS device to operate as a DHCPv6 client, we need to enable DHCPv6 client on a VLAN interface to obtain an IPv6 address from the DHCPv6 server in the network.

ExtremeXOS DHCPv6 client supports only the stateful DHCPv6 mode of operation in this release. This means an external DHCPv6 server is required to configure the IPv6 address and its related configurations.

Note
Stateless Auto configuration and stateless DHCPv6 modes are not supported in this release.

The mode of operation of the DHCPv6 client is based on the autonomous (A), managed (M) and other configuration (O) flags in the received router advertisement (RA) messages. If the managed bit is 1 and the other configuration bit is 0, the DHCPv6 client will act as a stateful client. In stateful mode, the client receives IPv6 addresses from the DHCPv6 server, based on the identity association for nontemporary addresses (IA_NA) assignment.

In other words ExtremeXOS DHCPv6 client sends IA_NA option as part of the DHCPv6 message.

Note
Identity association for temporary address (IA_TA) and Identity association for prefix delegation (IA_PD) is not supported in this release.

If the incoming RA points to act the interface in stateless DHCPv6 mode (managed bit is 0 and the other configuration bit is 1) or DHCPv6 auto configuration mode (autonomous bit is 1 and managed, other configuration bit is 0) ExtremeXOS will release the assigned (if already assigned using stateful DHCPv6 client mode) DHCPv6 IP and other parameters for this VLAN interface on which this RA is received.

This is because the mode of operation needs to be changed because ExtremeXOS did not support stateless DHCPv6 and auto configuration DHCPv6 client modes and there will not be any subsequent operation. In the stateful DHCPv6 client mode, the DHCPv6 client requests global addresses from the DHCPv6 server. Based on the DHCPv6 server's response, the DHCPv6 client assigns the global addresses to interfaces and sets a lease time for all valid responses. When the lease time expires, the DHCPv6 client renews the lease from the DHCPv6 server. To configure stateful DHCPv6 client based on identity association of non-temporary (IA_NA) address assignment for a VLAN interface use the following command.

```
enable dhcp ipv6 vlan vlan_name
```

To unconfigure the stateful DHCPv6 client for an interface use:

```
disable dhcp ipv6 vlan vlan_name
```

To display the current status of the DHCPv6 client VLAN state use:

```
show dhcp-client ipv6 state {vlan}
```

DHCPv6 client is not supported on VR-MGMT.

DHCP Unique Identifier

Each *DHCP* client and server has a DHCP unique identifier (DUID). A DHCP server uses DUIDs to identify clients for the selection of configuration parameters. DHCP clients use DUIDs to identify a server in messages where a server needs to be identified. Clients and servers treat DUIDs as opaque values and use it compare for equality.

The DUID is not required in all the DHCP messages. It has to be unique across all DHCP clients and servers. It has to be stable for any specific client or server - for example, a device's DUID should not change as a result of a change in the *VLAN*'s port.

DHCP Unique Identifier Content

A *DHCP* unique identifier (DUID) consists of a two-octet type code followed by a variable number of octets that make up the actual identifier. A DUID can be no more than 128 octets long (not including the type code). The following types are currently defined:

- Link-layer address plus time
- Vendor-assigned unique ID based on Enterprise Number
- Link-layer address

ExtremeXOS uses Link-layer address plus time as its default client DUID. To configure DHCPv6 client identifier type for the client use:

```
configure dhcp ipv6 client identifier-type [ link-layer | link-layer-
plus-time | vendor-specific ]
```

Client Requested DHCPv6 Options

The following default options are encapsulated by the DHCPv6 client when requesting DHCPv6 server.

1. OPTION_CLIENTID (code 1). The Client Identifier option is used to carry a DUID identifying a client between a client and a server.
2. OPTION_SERVERID (code 2). The Server Identifier option is used to carry a DUID identifying a server between a client and a server.

   Note
   OPTION_SERVERID option is included if DHCPv6 client is responding to a particular server.

3. OPTION_IA_NA (code 3). The client uses IA_NA options to request the assignment of non-temporary address assignment.
4. OPTION_IAADDR (code 5). The IA_ADDR is used to specify the IPv6 addresses associated with IA_NA or IA_TA option.

   Note
   OPTION_IAADDR is included only if the client lease is present.

Note
IAID is the last 4 bytes of the hardware address

5. OPTION_ORO (code 6). The Option Request option is used to identify a list of options in a message between client and server. A client includes an Option Request option in a Solicit, Request, Renew, Rebind, Confirm or Information-request message to inform the server about options the client is interested in.

The following are the Default requested options included by the ExtremeXOS DHCPv6 client as part of Option Request Option (OPTION_ORO) option:

1. OPTION_DNS_SERVERS (code 23).
2. OPTION_DOMAIN_LIST (code 24). Reference for above ORO options: http://tools.ietf.org/html/rfc3646.
3. OPTION_ELAPSED_TIME(code 8). A client MUST include an Elapsed Time option in messages to indicate how long the client has been trying to complete a *DHCP* message exchange.

Configuring DHCPv6 BOOTP Relay

To configure the relay function:

1. Configure *VLAN*s and IP unicast routing.

Note
As in DHCPv4, when you create an IPV6 VLAN interface, the corresponding disabledV6VlanList has an entry. The VLAN interface entry is removed whenever the bootpRelayv6 for the respective VLAN is enabled, and vice versa.

2. Enable the *DHCP* or BOOTP relay function using the following commands:

```
enable bootprelay {ipv4|ipv6} {vlan[vlan_name] |{vr} vr_name}|all
[{vr} vr_name]}
enable bootprelay {vlan[vlan_name] |{vr} vr_name}|all [{vr} vr_name]}
```

3. Configure the addresses to which DHCP or BOOTP requests should be directed using the following command:

```
configure bootreplay {ipv4} | {{vlan}vlan_name} [[add ip_address] |
delete [ip_address | all]]] | ipv6 [[add ipv6_address] | [delete
ipv6_address | all]]]]] {vr vrid}
configure bootprelay {ipv4 | ipv6} add ip_address {vr vrid}
```

Note
Use the `configure bootprelay ipv6` **`option interface-id`** *`InterfaceIDName`* **`vlan`** {*`vlan_name`*} command to set up a unique interface-id. It can be MAC-ID, or port-vlan combination. You can also use this command to set up dhcpv6 server/next hop for each VLAN interface, or across VR. A configuration applied to the VR level is populated to all VLAN V6 interfaces.

4. To delete an entry, use the following command:

   ```
   configure bootprelay {ipv4 | ipv6} delete [ip_address | all] {vr vrid}
   ```

5. To disable BOOTP relay on one or more VLANs, use the following command

   ```
   disable bootprelay {ipv4 | ipv6} {vlan [vlan_name] | {{vr} vr_name} | all [{vr} vr_name]}
   disable bootprelay {{vlan} [vlan_name] | {{vr} vr_name} | all [{vr} vr_name]}
   ```

   Note
   When *VRRP (Virtual Router Redundancy Protocol)* and BOOTP/DHCP relay are both enabled on the switch, the relayed BOOTP agent IP address is the actual switch IP address, not the virtual IP address.

Configure Route Compression

This helps with route optimization and scaling. This feature allows you to install only less specific routes in the table when overlapping routes with the same nexthop exist. For detailed information about route compression, see Hardware Routing Table Management.

Note
This feature is enabled by default starting from ExtremeXOS 15.6.

- Enable this feature.

  ```
  enable iproute ipv6 compression {vr vrname}
  ```

- Disable this feature.

  ```
  disable iproute ipv6 compression {vr vrname}
  ```

Hardware Forwarding Behavior

Table 63 shows the Extreme switch platforms and the ExtremeXOS software versions that provide hardware forwarding support for IPv6 unicast features.

Table 63: IPv6 Unicast Features

Switch Model	ExtremeXOS Release	Features
4120	32.7 and later	IPv6 Unicast forwarding
4220	32.7 and later	IPv6 Unicast forwarding
5120	33.3.1 and later	IPv6 Unicast forwarding
5320	ExtremeXOS 31.6 and later	IPv6 Unicast forwarding IPv6 tunneling
5420	ExtremeXOS 31.3 and later	IPv6 Unicast forwarding IPv6 tunneling
5520	ExtremeXOS 31.1 and later	IPv6 Unicast forwarding IPv6 tunneling

Table 63: IPv6 Unicast Features (continued)

Switch Model	ExtremeXOS Release	Features
5720	Switch Engine 32.1 and later	IPv6 Unicast forwarding IPv6 tunneling
7520	Switch Engine 32.4 and later	IPv6 Unicast forwarding IPv6 tunneling
7720	Switch Engine 32.4 and later	IPv6 Unicast forwarding IPv6 tunneling

Hardware Forwarding Limitations

All switches can use hardware forwarding when the route mask is 64 bits or less. If the route mask is greater that 64 bits, limitations apply based on the hardware platform.

This support was added in ExtremeXOS Release 12.4 by using some of the slices previously used for *ACL* support to create a Greater Than 64 Bit (GT64B) table. The GT64B table stores only those routes with a mask greater than 64 bits. When IPv6 forwarding is enabled, the switch behavior is as follows:

- Fewer slices are available for ACLs.

  Use show access-list usage acl-slice to check if any slice is unused. If no slice is available, consider disabling a feature that is consuming ACL slices if that feature is not required. Features that are enabled by default such as *IGMP (Internet Group Management Protocol)* Snooping or MLD Snooping can be disabled to free up ACL resources if not required.
- Table-full messages appear when there is no more space in the GT64B table.
- If an eligible route cannot be added to the GT64B table (because the table is full), there is no guarantee that traffic for that route will be properly routed.
- If enabled, route compression for IPv6 can make room for additional routes by reducing the number of entries in the GT64B table.
- When an IPv6 address with a mask greater that 64 bits is configured on a *VLAN* or tunnel, that address is automatically added to the GT64B table.

All switches support hardware forwarding routes with masks greater than 64 bits. For number of routes supported, see the limits for the particular switch in the *ExtremeXOS Release Notes*.

Starting with ExtremeXOS 22.5, the ipv6-mask-length option provides greater hardware route scale and IP route sharing (*ECMP*) support for IPv6 “long-mask routes”, meaning IPv6 subnets with mask lengths 65–128 bits. This provides additional scale and resilience for IPv6 host routes whose mask length is 128 bits. Increasing scale and providing ECMP for IPv6 mask 65–128 routes decreases IPv4 route scale. The default IPv6 mask length is 64.

The `configure forwarding internal-tables [ **l2-and-l3** | **more** [**l2** | **l3-and-ipmc** | **routes** {**ipv6-mask-length** [**64** | **128**]}]]` command provides the ability to support additional L2 and L3 hosts, routes, or multicast table entries.

This support was added in ExtremeXOS Release 12.4 by using a hardware table designed for this purpose. When IPv6 forwarding is enabled, the switch behavior is as follows:

- If no space is available in the hardware table, there is no guarantee that traffic for that route will be properly routed.
- If enabled, route compression for IPv6 can make room for additional routes by reducing the number of entries in the hardware table.
- When an IPv6 address with a mask greater that 64 bits is configured on a VLAN or tunnel, that address is automatically added to the hardware table.

Hardware Tunnel Support

The platforms and software versions that provide hardware forwarding support for IPv6 traffic for both IPv6-in-IPv4 and 6to4 tunnels for various platforms are shown in the following table. In all the other platforms, tunnel traffic is forwarded in software and the statistics can be viewed from the show ipstats ipv6 command.

Note

The MTU for IPv6 Tunnels is set to 1480 bytes, and is not user-configurable. Configuring jumbo frames has no effect on the MTU size for IPv6 tunnels.

Routing Configuration Example

Figure 106 on page 985 illustrates a switch with three VLANs defined as follows:

- Finance
  - All ports on slots 1 and 3 have been assigned.
  - IP address 192.207.35.1.
- Personnel
  - Protocol-sensitive *VLAN* using the IP protocol.
  - All ports on slots 2 and 4 have been assigned.
  - IP address 192.207.36.1.
- MyCompany
  - Port-based VLAN.
  - All ports on slots 1 through 4 have been assigned.

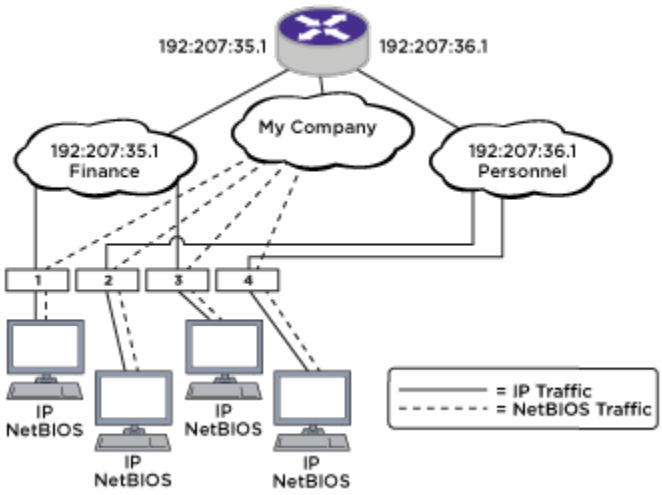


Figure 106: Unicast Routing Configuration Example

The stations connected to the system generate a combination of IP traffic and NetBIOS traffic. The IP traffic is filtered by the protocol-sensitive VLANs. All other traffic is directed to the VLAN MyCompany.

In this configuration, all IP traffic from stations connected to slots 1 and 3 have access to the router by way of the VLAN Finance. Ports on slots 2 and 4 reach the router by way of the VLAN Personnel. All other traffic (NetBIOS) is part of the VLAN MyCompany.

The example in the above figure is configured as follows:

```
create vlan Finance tag 10
create vlan Personnel tag 11
create vlan MyCompany

configure Finance protocol ip
configure Personnel protocol ip

configure Finance add port 1:*,3:* tag
configure Personnel add port 2:*,4:* tag
configure MyCompany add port all

configure Finance ipaddress 192.207.35.1
configure Personnel ipaddress 192.207.36.1
configure rip add vlan Finance
configure rip add vlan Personnel

enable ipforwarding
enable rip
```

Tunnel Configuration Examples

6in4 Tunnel Configuration Example

Figure 107 illustrates a 6in4 tunnel configured between two IPv6 regions across an IPv4 region.

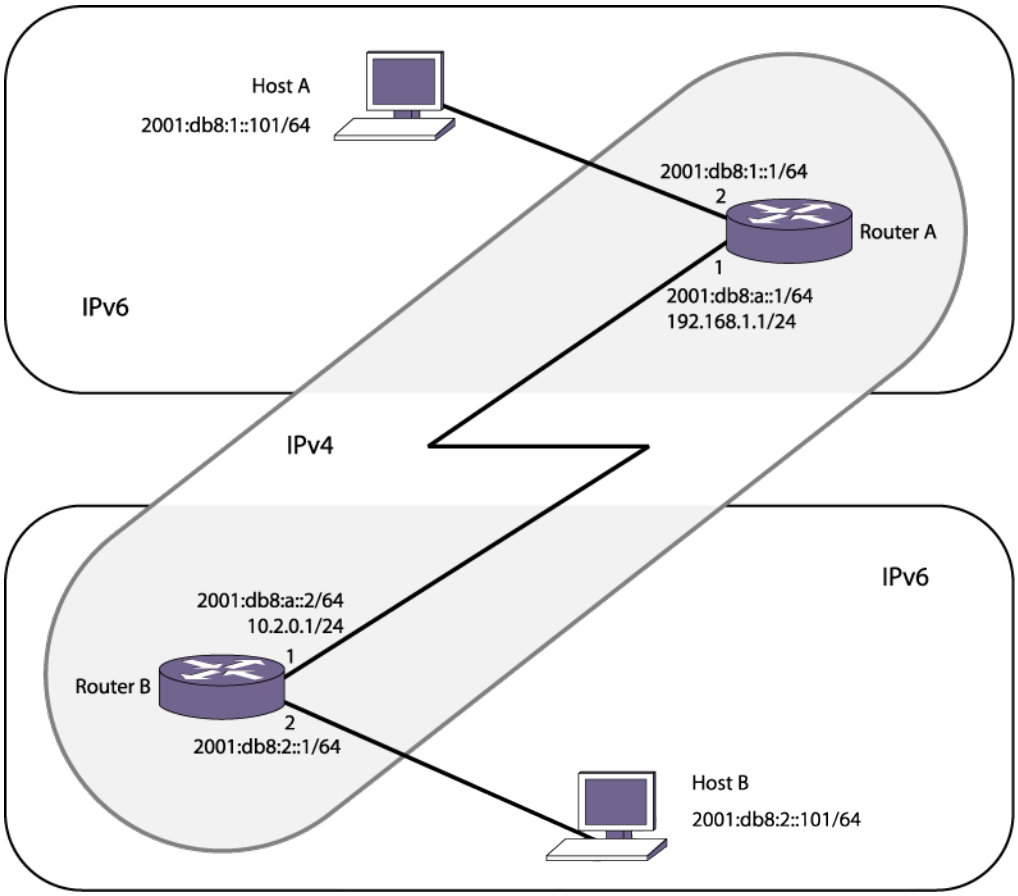


Figure 107: 6in4 Tunnel Example

Router A has an interface to an IPv4 region with the address 192.168.1.1 (for this example we are using private IPv4 addresses, but to tunnel across the Internet, you would use a public address). Router B has an IPv4 interface of 10.2.0.1. The IPv4 interface must be created before the tunnel is configured and cannot be deleted until the tunnel is deleted.

This example has one subnet in each IPv6 region, 2001:db8:1::/64 for Router A and 2001:db8:2::/64 for Router B. Hosts A and B are configured to use IPv6 addresses 2001:db8:1::101 and 2001:db8:2::101 respectively.

For traffic to move from one region to the other, there must be a route. In this example, a static route is created, but you could enable *RIPng* or *OSPFv3* on the tunnel interface.

In this example, we assume that the IPv4 network can route from Router A to Router B (in other words, some IPv4 routing protocol is running on the public-ipv4 interfaces). For platforms on which hardware based tunneling is supported (See the following table), IPv4 forwarding needs to be enabled on the tunnel source *VLAN*. However, in platforms on which IPv6-in-IPv4 tunnels are supported in software only, you do not need to enable IPv4 forwarding on the public interfaces in this example unless you are also routing IPv4 traffic on them (in this example, it is assumed you are running no IPv4 traffic inside your respective IPv6 networks, although you could).

When Host A needs to send a packet to 2001:db8:2::101 (Host B), it forwards it to Router A. Router A receives an IPv6 packet from the IPv6 source address 2001:db8:1::101 to the destination 2001:db8:2::101. Router A has the static route, for the route 2001:db8:2::/64 with next hop 2001:db8:a::2 (Router B) through the tunnel interface. So Router A encapsulates the IPv6 packet inside an IPv4 header with the source address 192.168.1.1 and destination address 10.2.0.1. The encapsulated IPv6 packet passes through the IPv4 network and reaches the other end of the tunnel—Router B. Router B decapsulates the packet and removes the IPv4 header. Router B then forwards the IPv6 packet to the destination host—Host B.

Note
Each IPv6 packet is encapsulated inside an IPv4 header (20 bytes) before it is forwarded via a IPv6-in-IPv4 tunnel. For example, a 66-byte packet from Host A will be encapsulated and forwarded as a 86-byte packet by Router A.

Router A

```
configure vlan default delete port all
create vlan public-ipv4
configure vlan public-ipv4 add port 1 untagged
configure vlan public-ipv4 ipaddress 192.168.1.1/24
create tunnel public6in4 ipv6-in-ipv4 destination 10.2.0.1 source 192.168.1.1
configure tunnel public6in4 ipaddress 2001:db8:a::1/64
enable ipforwarding ipv6 public6in4
create vlan private-ipv6
configure vlan private-ipv6 add port 2 untagged
configure vlan private-ipv6 ipaddress 2001:db8:1::1/64
enable ipforwarding ipv6 private-ipv6
configure iproute add 2001:db8:2::/64 2001:db8:a::2
enable ipforwarding public-ipv4
```

Router B

```
configure vlan default delete port all
create vlan public-ipv4
configure vlan public-ipv4 add port 1 untagged
configure vlan public-ipv4 ipaddress 10.2.0.1/24
create tunnel public6in4 ipv6-in-ipv4 destination 192.168.1.1 source 10.2.0.1
configure tunnel public6in4 ipaddress 2001:db8:a::2/64
enable ipforwarding ipv6 public6in4
create vlan private-ipv6
configure vlan private-ipv6 add port 2 untagged
configure vlan private-ipv6 ipaddress 2001:db8:2::1/64
enable ipforwarding ipv6 private-ipv6
configure iproute add 2001:db8:1::/64 2001:db8:a::1
enable ipforwarding public-ipv4
```

6to4 Tunnel Configuration Example

Figure 108 illustrates a 6to4 tunnel configured between two IPv6 regions across an IPv4 region.

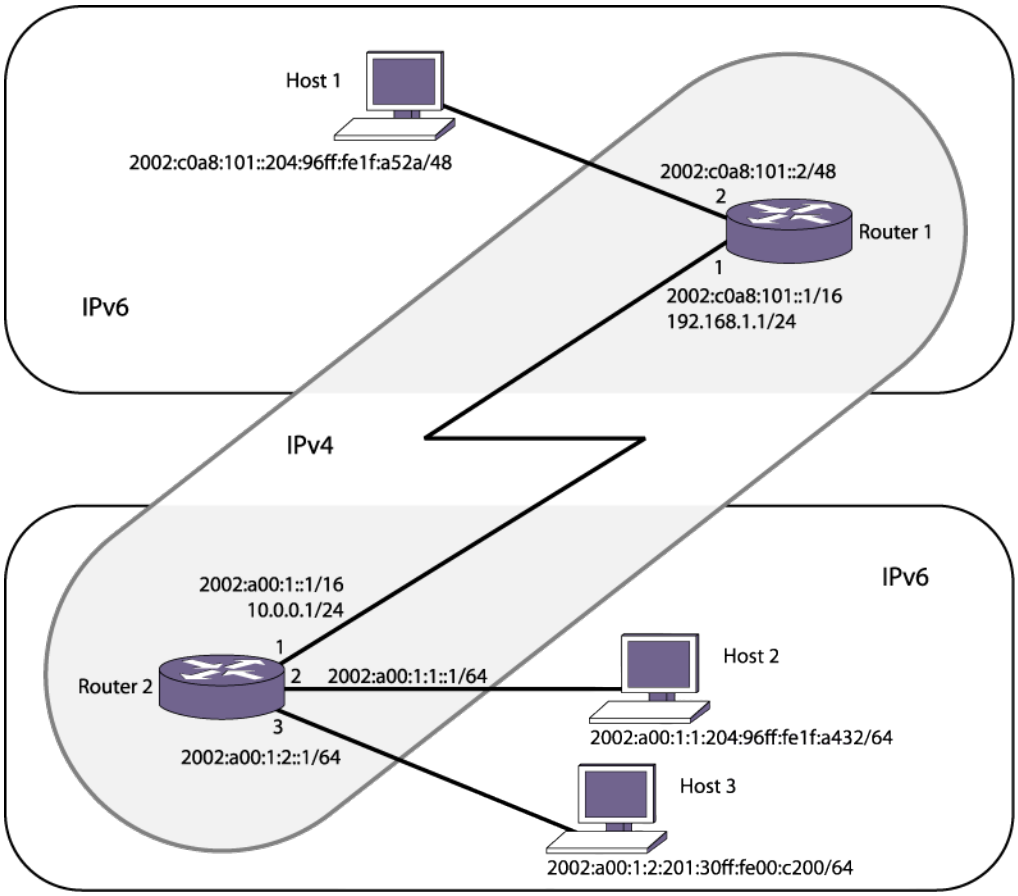


Figure 108: 6to4 Tunnel Configuration Example

Router 1 has an interface to an IPv4 region with the address 192.168.1.1 (for this example we are using private IPv4 addresses, but to tunnel across the Internet, you would use a public address). Router 2 has an IPv4 interface of 10.0.0.1. The IPv4 interface must be created before the tunnel is configured and cannot be deleted until the tunnel is deleted.

The IPv6 endpoints of 6to4 tunnels must follow the standard 6to4 address requirement. The address must be of the form `2002:<IPv4_source_endpoint>::/16`, where <IPv4_source_endpoint> is replaced by the IPv4 source address of the endpoint, in hexadecimal, colon separated form. For example, for a tunnel endpoint located at IPv4 address 10.20.30.40, the tunnel address would be 2002:0a14:1e28::/16. In hex, 10 is 0a, 20 is 14, 30 is 1e and 40 is 28.

This example shows a simple setup on the Router 1 side (one big /48 IPv6 routing domain with no subnets), and a slightly more complex setup on the Router 2 side (two subnets :0001: and :0002: that are /64 in length). Hosts 1, 2, and 3 will communicate using their global 2002: addresses.

The hosts in this example configure themselves using the EUI64 interface identifier derived from their MAC addresses. Refer to your host OS vendor's documentation for configuring IPv6 addresses and routes.

In this example, we assume that the IPv4 network can route from Router 1 to Router 2 (in other words, some IPv4 routing protocol is running on the public-ipv4 interfaces). However, you do not need to enable IPv4 forwarding on the public interfaces in this example unless you are also routing IPv4 traffic on them (in this example, it is assumed you are running no IPv4 traffic inside your respective IPv6 networks, although you could).

Router 1

```
configure vlan default delete port all
create vlan public-ipv4
configure vlan public-ipv4 add port 1 untagged
configure vlan public-ipv4 ipaddress 192.168.1.1/24
create tunnel public6to4 6to4 source 192.168.1.1
configure tunnel public6to4 ipaddress 2002:c0a8:0101::1/16
enable ipforwarding ipv6 public6to4
create vlan private-ipv6
configure vlan private-ipv6 add port 2 untagged
configure vlan private-ipv6 ipaddress 2002:c0a8:0101::2/48
enable ipforwarding ipv6 private-ipv6
```

Router 2

```
configure vlan default delete port all
create vlan public-ipv4
configure vlan public-ipv4 add port 1 untagged
configure vlan public-ipv4 ipaddress 10.0.0.1/24
create tunnel public6to4 6to4 source 10.0.0.1
configure tunnel public6to4 ipaddress 2002:0a00:0001::1/16
enable ipforwarding ipv6 public6to4
create vlan private-ipv6-sub1
configure vlan private-ipv6-sub1 add port 2 untagged
configure vlan private-ipv6-sub1 ipaddress 2002:0a00:0001:0001::1/64
enable ipforwarding ipv6 private-ipv6-sub1
create vlan private-ipv6-sub2
configure vlan private-ipv6-sub2 add port 3 untagged
configure vlan private-ipv6-sub2 ipaddress 2002:0a00:0001:0002::1/64
enable ipforwarding ipv6 private-ipv6-sub2
```

Host Configurations

The IPv6 addresses of these hosts are based on their MAC address-derived EUI64 interface identifiers and the address prefixes for their subnets. Each host must also have a static route configured on it for 6to4 addresses.

Host 1:

- MAC address—00:04:96:1F:A5:2A
- IPv6 address—2002:c0a8:0101::0204:96ff:fe1f:a52a/48
- Static route—destination 2002::/16, gateway 2002:c0a8:0101::2

Host 2:

- MAC address—00:04:96:1F:A4:32
- IP address—2002:0a00:0001:0001:0204:96ff:fe1f:a432/64
- Static route—destination 2002::/16, gateway 2002:0a00:0001:0001::1

Host 3:

- MAC address—00:01:30:00:C2:00
- IP address—2002:0a00:0001:0002:0201:30ff:fe00:c200/64
- Static route—destination 2002::/16, gateway 2002:0a00:0001:0002::1

GRE Tunnel Configuration Example

Router 1 Configuration

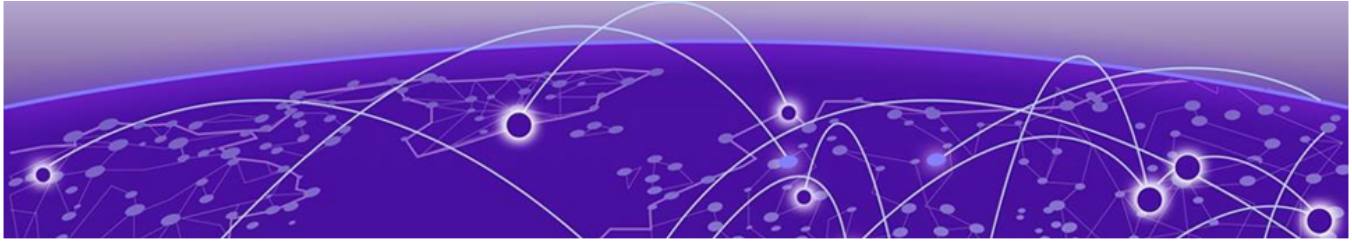
```
unconfigure switch all
configure default del port all
create vlan inet
configure vlan inet add port 24
configure vlan inet ipa 1.1.1.1/24
create vlan users
configure vlan users add port 1
configure vlan users ipa 100.0.0.1/24
create tunnel mytunnel gre destination 1.1.1.2 source 1.1.1.1
configure tunnel "mytunnel" ipaddress 2.0.0.1/24
configure iproute add 200.0.0.0/24 2.0.0.2
enable ipforwarding
```

Router 2 Configuration

```
create vlan inet
configure vlan inet add port 10:1
configure vlan inet ipa 1.1.1.2/24
create vlan users
configure vlan users add port 10:2
configure vlan users ipa 200.0.0.1/24
create tunnel mytunnel gre destination 1.1.1.1 source 1.1.1.2
configure tunnel "mytunnel" ipaddress 2.0.0.2/24
configure iproute add 100.0.0.0/24 2.0.0.1
enable ipforwarding

Host IP connected to router 1-100.0.0.1
Host IP connected to router 2-200.0.0.1
```

Note

If a transit node is present between the tunneling sites, you should configure static or enable IBGP between the sites for network connectivity.



IS-IS

IS-IS Overview on page 992
Route Redistribution on page 998
Configuring IS-IS on page 1000
Displaying IS-IS Information on page 1005
Managing IS-IS on page 1006
Configuration Example on page 1012

Note
The IS-IS feature is supported only at and above the license level listed for this feature in the license tables in the licensing guide.

This chapter introduces IS-IS, a link state protocol that distributes routing information between routers belonging to an autonomous system. It provides configuration commands and examples and information about configuring, displaying, and managing IS-IS on a network. This chapter assumes that you are already familiar with Intermediate System-Intermediate System (IS-IS) and IP routing.

If not, refer to the following publications for additional information:

- RFC 1195—Use of OSI IS-IS for Routing in TCP/IP and Dual Environments
- RFC 2763—Dynamic Hostname Exchange Mechanism for IS-IS
- RFC 2966—Domain-Wide Prefix Distribution with Two-Level IS-IS
- RFC 2973—IS-IS Mesh Groups
- RFC 3373—Three-Way Handshaking for IS-IS Point-to-Point Adjacencies
- RFC 3719—Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)
- RFC 3787—Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)
- draft-ieft-isis-ipv6-06—Routing IPv6 with IS-IS
- draft-ietf-isis-restart-02—Restart signaling for IS-IS
- draft-ietf-isis-wg-multi-topology-11—Multi-topology (MT) routing in IS-IS
- ISO 10589—OSI IS-IS Intra-Domain Routing Protocol (also available as RFC 1142)
- *Interconnections: Bridges and Routers* by Radia Perlman, ISBN 0-201-56332-0, Published by Addison-Wesley Publishing Company

IS-IS Overview

IS-IS is a link state protocol that distributes routing information between routers belonging to a single IP domain; the IP domain is also known as an autonomous system (AS). In a link-state routing protocol, each router maintains a database describing the topology of the AS. Each participating router has an identical database maintained from the perspective of that router.

From the link state database (LSDB), each router constructs a tree of shortest paths, using itself as the root. The shortest path tree provides the route to each destination in the AS. When several equal-cost routes to a destination exist, traffic can be distributed among them. The cost of a route is described by a single metric.

IS-IS is an interior gateway protocol (IGP), as are *RIP (Routing Information Protocol)* and *OSPF (Open Shortest Path First)*. Unlike RIP and OSPF, IS-IS was not initially designed for IP. RFC 1195 specifies how IS-IS can run in an IP environment. The Extreme Networks implementation supports IS-IS only in IP environments. RIP, OSPF, and IS-IS are compared in RIP. The IPv6 versions of these protocols are compared in RIPng.

Establishing Adjacencies

An adjacency is an acknowledged relationship between two IS-IS routers. An adjacency must be established before two IS-IS routers can exchange routing information.

IS-IS routers establish adjacencies by exchanging hello PDUs, which are also called Intermediate System to Intermediate System Hellos (IIHs). Hello PDUs contain some interface configuration and capability information. Once a pair of neighbors exchanges hello PDUs with acceptably matching configuration and capabilities, an adjacency is formed. Hello PDUs are sent periodically by each party to maintain the adjacency.

After an adjacency is formed, information about the adjacency is stored in a link state PDU (LSP), which is stored in the router link state database (LSDB). Routers periodically flood all of their LSPs to all other network nodes. When a router receives LSPs, it adds the LSPs to its LSDB, and uses the LSDB to calculate routes to other routers. These database maintenance operations are performed a little differently for the two adjacency types, point-to-point, and broadcast.

Point-to-Point Adjacency

Point-to-point adjacencies can include no more than two routers in the same *VLAN (Virtual LAN)*. Figure 109 shows a point-to-point adjacency.

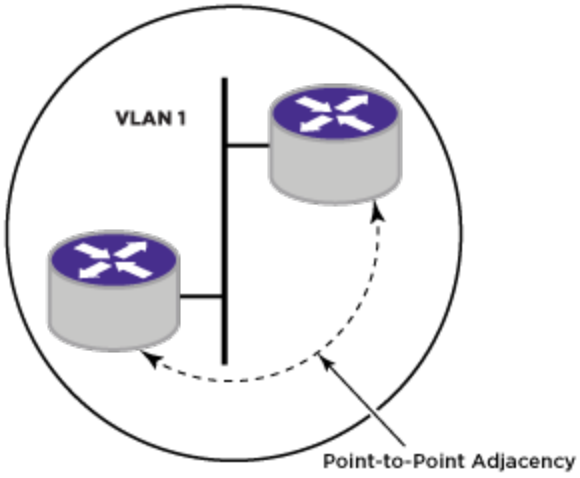


Figure 109: Point-to-Point Adjacency

Once a point-to-point adjacency is established, each router sends a CSNP (Complete Sequence Number PDU) listing a summary of all its LSPs. When a router receives its neighbor's CSNP, it sends any LSP it has that is either not present in the CSNP or is newer than the version in the CSNP. In a point-to-point adjacency, partial sequence number PDUs (PSNPs) are used to acknowledge each LSP a router receives from its neighbor. If a PSNP is not received within a configurable period of time, unacknowledged LSPs are re-sent.

A disadvantage to point-to-point adjacencies is that they do not scale well. Figure 110 shows a four-router network with point-to-point adjacencies.

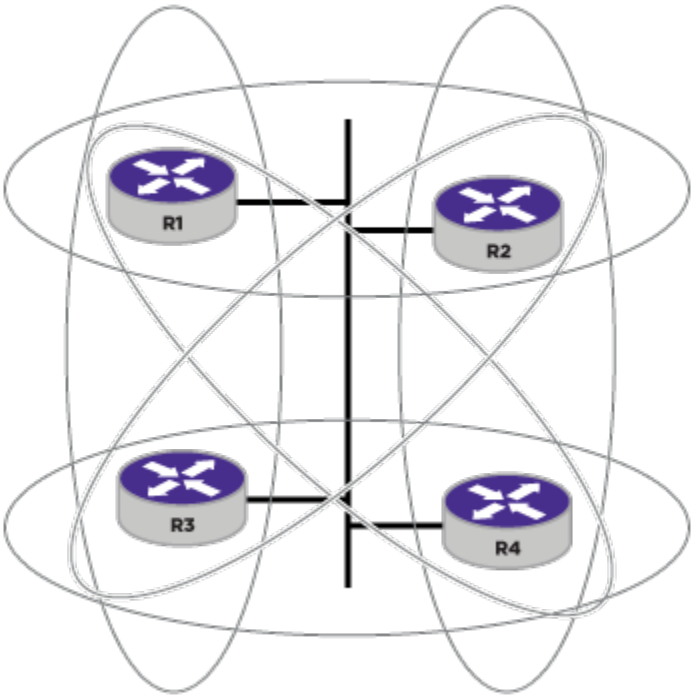


Figure 110: Point-to-Point Adjacencies in a Four-Router Network

In this network, each of the ellipses represents a point-to-point adjacency. Each of the four routers periodically sends all of its LSPs to the other three routers. Each, in turn, will flood the received LSPs to the other two since they have no way of knowing which routers have already received them, generating N2 LSPs. This network routing traffic reduces the bandwidth available for data traffic. For networks that only have two routers and are not likely to grow, a point-to-point adjacency is appropriate.

Broadcast Adjacency

Broadcast adjacencies can include two or more routers in a single adjacency. Figure 111 depicts a broadcast adjacency.

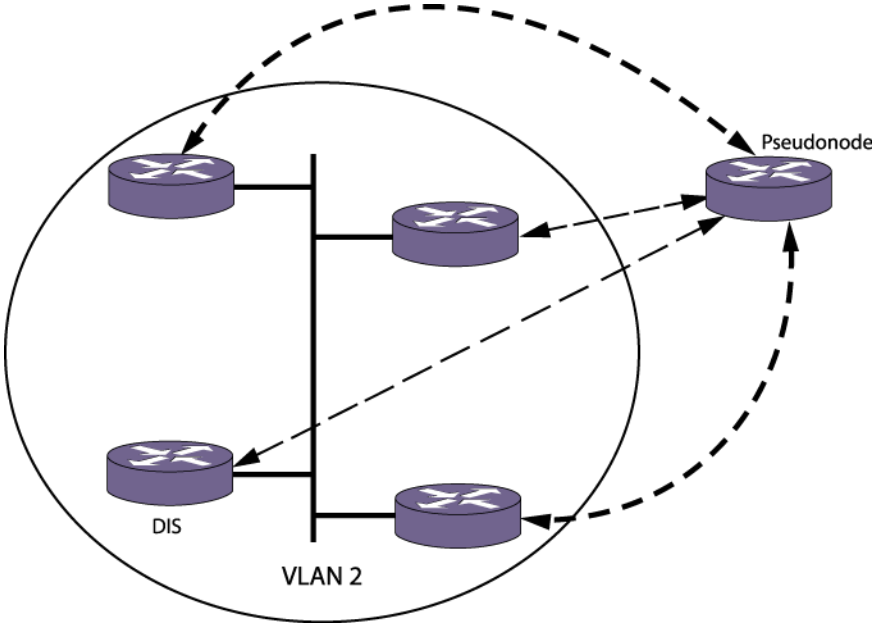


Figure 111: Broadcast Adjacencies in a Four-Router Network

When broadcast adjacencies are formed, one of the routers participating in the adjacency is elected designated intermediate system (DIS). The election is determined by a DIS priority configured for each router. In the event of a tie, the router with the numerically highest MAC address wins.

A broadcast network can be considered a virtual node, or pseudonode, to which all routers have a zero-cost adjacency.

The DIS acts on behalf of the pseudonode by advertising an LSP listing all routers in the adjacency with zero-cost metric. The DIS also periodically sends a complete sequence number PDU (CSNP), which lists all LSPs in the link-state database. If a router sees that it is missing one or more of the entries, it multicasts a request for them using a PSNP. Only the DIS responds to this request by sending the requested LSPs. All routers multicast their originated LSPs as they are refreshed, and multicast periodic hellos to the network.

The default configuration creates broadcast adjacencies.

IS-IS Hierarchy

IS-IS has a two-level hierarchy. IS-IS routers may participate in either level or both. Figure 112 shows a basic IS-IS AS.

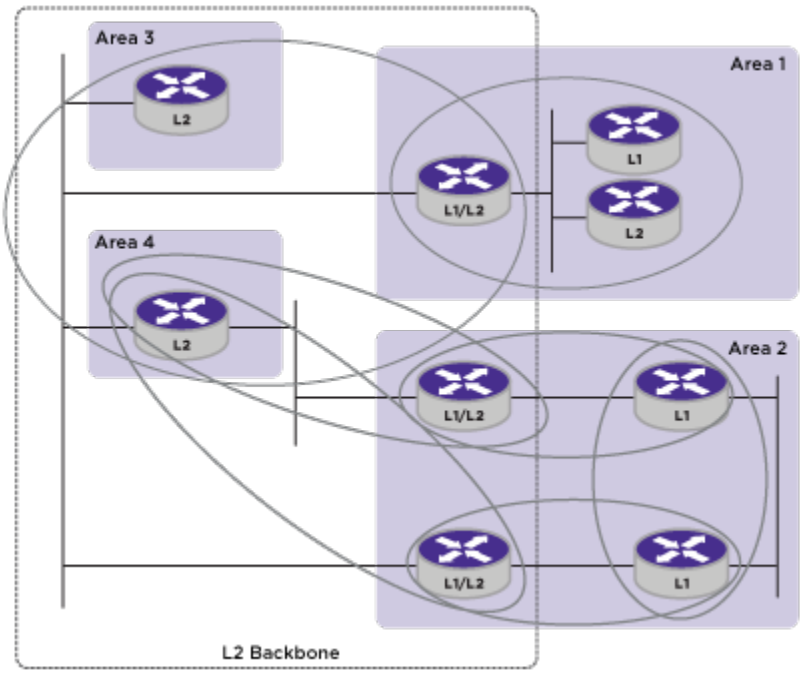


Figure 112: Basic Autonomous System

In the network in the above figure, each of the ellipses represents an adjacency. Level 1 (L1) routers within an area need only know the topology within that area as well as the location of the nearest L1/L2 attached router. An attached router is one that participates in both levels and can reach at least one other L1/L2 or L2 router with a different area address.

In an AS, there is only one L2 area, and it serves as a backbone area between L1 areas. This L2 area is also called a domain (not to be confused with the entire AS routing domain). L2 routers communicate with one another irrespective of L1 area membership or L2 area address.

Note
On any switch, the maximum number of L1/L2 adjacencies is one half the maximum number of L1 or L2 adjacencies because an L1/L2 adjacency defines both an L1 adjacency and an L2 adjacency. The supported adjacency limits are defined in the *ExtremeXOS Release Notes*.

For information on creating and managing IS-IS areas, see Managing an IS-IS Area Address on page 1009.

IS-IS and IP Routing

IS-IS is not inherently an IP Routing protocol. Its control packets do not use IP. RFC 1195 specifies the use of IS-IS for IP Routing.

The ExtremeXOS software implementation of IS-IS requires that at least one IPv4 or IPv6 address be assigned to an interface.

IP addresses and subnets can be assigned independent of area structure provided that neighboring interfaces are on the same subnet. L1 routers exchange directly reachable IP address/mask/metric tuples. When routing in an L1 area, packets are routed via L1 if the destination address is reachable within the area. Otherwise packets are forwarded to the nearest L2 router. L2 routers advertise all IP addresses reachable in their L1 area (if they are a member of one), as well as directly reachable addresses.

Summary Addresses

Routers can be manually configured to advertise abbreviated versions of reachable addresses, which are called summary addresses. By aggregating many routes into one summary address, route computation and lookup can be made more efficient. If packets arrive at the router as a result of the summary address but have no actual route (as a result of the summary address covering more address space than is actually used), the packets are discarded.

For instructions on managing summary addresses, see Managing IP Summary Addresses on page 1008.

External Connectivity

Externally reachable IP addresses are those learned or exported from other routing protocols like *RIP*, *BGP (Border Gateway Protocol)*, and *OSPF*. When using narrow metrics, external routes may use internal or external metrics. When calculating SPF, routes with internal metrics are preferred over routes with external metrics. When external metrics are used to compare routes, the internal cost is not considered unless the external metrics are equal.

Authentication

Entire packets (as opposed to individual TLVs in a packet) can be authenticated. If authentication fails on a packet, the entire packet is discarded. Multi-part packets are authenticated separately. Routers are not required to support authentication. The ExtremeXOS software provides optional support for plain-text authentication.

Dynamic Hostname

The dynamic hostname exchange extension helps address the usability issues of maintaining IS-IS routers. IS-IS system IDs (which default to the switch MAC address) are not very readable. When the dynamic hostname feature is enabled, learned IS-IS hostnames are used in place of system IDs in log entries where possible. In addition, hostname TLVs appear in the `show isis lsdb detail` command display.

For instructions on managing the dynamic hostname feature, see Configuring the Dynamic Hostname Feature on page 1004.

Route Leaking

Route leaking allows L2 LSPs to be sent into L1 areas. Route leaking is configured with the command:

```
configure isis area area_name interlevel-filter level 2-to-1 [policy |
block-all | allow-all] {ipv4 | ipv6}
```

The supplied policy defines what route or routes should be leaked.

Caution
Route leaking can reduce scalability and performance.

Note
Tracert does not work if the VRF leaked route is one of the hop(s) to destination.

Metric Types

Interface metrics are available in two sizes: a 6-bit narrow metric and a 24-bit wide metric. By default only narrow metrics are used. Wide metrics allow for greater flexibility, however, and are required in order to use some extensions of IS-IS, such as IPv6. All routers in an IS-IS area must use the same metric style.

For instructions on configuring interface metric style and values, see Configuring VLAN Interface Metrics on page 1010. Refer to RFC 3787, Section 5.1, for migration details. Note that the ExtremeXOS software does not support section 5.2.

IS-IS Restart

When an IS-IS router restarts, neighbors time out adjacencies and the network converges around it. IS-IS restart support, with the help of restart helper neighbors, can prevent this. A restarting router can send a restart request to indicate to its neighbors that it is restarting. Neighbors—provided they are helpers—send the restarting router a CSNP so that it can reconstruct its LSDB by tracking which LSPs it has and has not received. Neighbors can still time out the adjacency, so this might not work in environments with low hold timers.

IS-IS restart can be configured for planned and unplanned events. Planned events are user-initiated process restarts and user-initiated failovers. Unplanned events are process restarts or failovers that are not administratively initiated.

For information on configuring IS-IS restart, see Configuring the Graceful Restart Feature.

IPv4 and IPv6 Topology Modes

Interfaces can be IS-IS-enabled for IPv4, IPv6, or both. Within an IS-IS area, IPv4 and IPv6 can be supported in a single topology or in a multiple topology. A transition configuration is provide for migrating between the single and multiple topologies.

In a single topology, all interfaces in an area must be configured for IPv4, IPv6, or both IPv4 and IPv6. Adjacencies are denied if the topology configuration between two routers does not match. A single SPF calculation is performed for each route in a single topology area.

In a multiple topology area, each interface can be configured for IPv4, IPv6, or both IPv4 and IPv6. The router creates separate topologies for IPv4 and IPv6. Adjacencies are permitted between interfaces with different configurations, provided that the neighbors support at least one common protocol. Separate SPF calculations are computed for IPv4 and IPv6 routes.

Note
Although the IPv4 and IPv6 topologies can be different when multiple topologies are enabled, the topologies must be convex. That is, no IPv4 interface can be reachable only through traversal of IPv6 interfaces, and vice versa. All routers in an area must use the same topology mode.

When the transition topology mode is configured, the router allows adjacencies between any two interfaces, and distributes both single topology and multiple topology TLVs. Transition mode permits both single and multiple topologies to coexist, but it generates more traffic. Transition mode should be used only for transitioning between modes and should be disabled after the transition is complete.

By default, a single topology is used for both IPv4 and IPv6. As a result, all interfaces must be homogeneously ISIS-enabled for IPv4, IPv6, or both.

To change the configured topology mode, see Configure the Multi-Topology Feature on page 1005.

Route Redistribution

More than one routing protocol can be enabled simultaneously on the switch.

Route redistribution allows the switch to exchange routes, including static routes, between the routing protocols. Figure 113 is an example of route redistribution between an OSPF AS and a RIP AS.

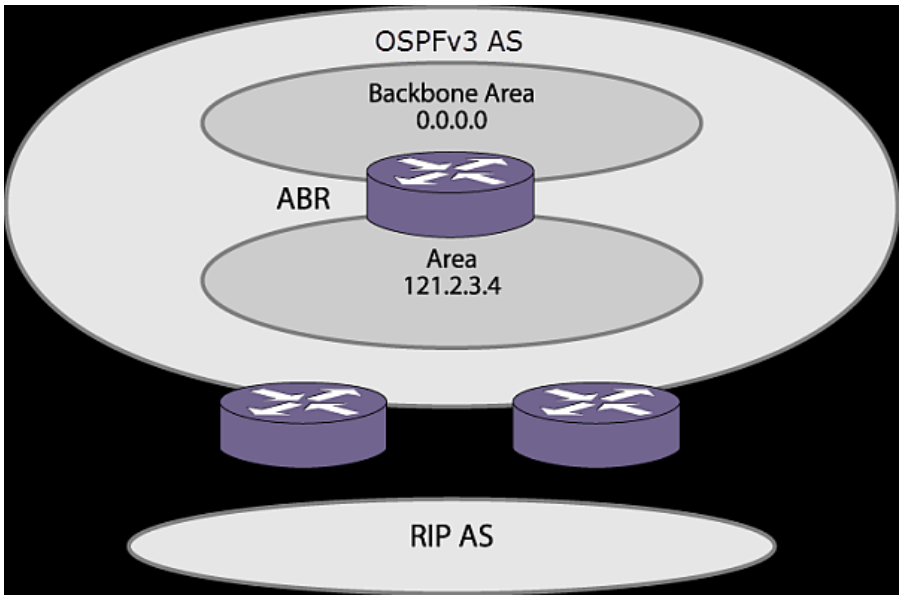


Figure 113: OSPF Route Redistribution

Configuring Route Redistribution

Exporting routes from one protocol to another and in the reverse direction are discrete configuration functions. For example, to run IS-IS and *RIP* simultaneously, you must first configure both protocols and then verify the independent operation of each. Then you can configure the routes to export from IS-IS to RIP and the routes to export from RIP to IS-IS. Likewise, for any other combinations of protocols, you must separately configure each to export routes to the other.

This section describes how to inject routing information learned from other IP routing protocols into an IS-IS domain, thereby advertising their reachability in the IS-IS domain. These routes are included in the locally originated LSPs. For information on exporting routes learned in IS-IS to another routing protocol, see the chapter that describes the destination routing protocol.

When you export routes from another protocol to IS-IS, the metric and the type of metric are configured. You can also configure a policy to filter out unwanted external routes.

- To enable or disable the exporting of *BGP*, *OSPF*, RIP, static, and direct (interface) IPv4 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export {ipv4} route-type [policy | metric
  mvalue {metric-type [internal | external]}] {level[1| 2 | both-1-
  and-2]}
  ```

  ```
  disable isis area area_name export {ipv4} route-type
  ```

- To enable or disable the exporting of *OSPFv3 (Open Shortest Path First version 3)*, *RIPng (Routing Information Protocol Next Generation)*, static, and direct (interface) IPv6 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export ipv6 route-type [policy | metric
  mvalue] {level[1| 2 | both-1-and-2]}
  ```

```
disable isis area area_name export ipv6 route-type
```

The cost metric is inserted for all *OSPF*, RIP, static, and direct routes injected into IS-IS.

The same cost and type values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route. The exported routes can also be filtered using policies.

- Verify the configuration using the command:

```
show isis area [area_name | all]
```

Configuring IS-IS

Configuring L1 Routers

To configure a switch to operate as a level 1 IS-IS router, do the following:

1. Prepare the IP interfaces that will support level 1 IS-IS routing as follows:
   a. Add an IPv4 or IPv6 address.
   b. Enable IP or IPv6 forwarding.
2. Create the IS-IS routing process, which is also called an area, using the following command:

   ```
   create isis area area_name
   ```
3. Configure the routing process to serve as a L1-only router using the following command:

   ```
   configure isis area area_name is-type level [1 | 2 | both-1-and-2]
   ```

   Specify 1 for the level option using the following command:
4. Add an IS-IS area level-1 address to the router using the following command:

   ```
   configure isis area area_name add area-address area_address
   ```
5. Add IS-IS-eligible interfaces to the area using the following command:

   ```
   configure isis add [vlan all | {vlan} vlan_name] area area_name {ipv4 | ipv6}
   ```

   An IS-IS-eligible interface is one that already has the appropriate IP address type (IPv4 or IPv6) address assigned to it.
6. The default IS-IS system ID is the switch MAC address. If you want to change the default IS-IS system ID, using the following command:

   ```
   configure isis area area_name system-id [automatic | system_id]
   ```
7. Enable the IS-IS router using the following command:

   ```
   enable isis {area area_name}
   ```

Configuring L1/L2 Routers

To configure a switch to operate as a L1/L2 IS-IS router, do the following:

1. Prepare the IP interfaces that will support level 1 IS-IS routing as follows:
   a. Add an IPv4 or IPv6 address.
   b. Enable IP or IPv6 forwarding.
2. Create the IS-IS routing process, which is also called an area, using the following command:

   ```
   create isis area area_name
   ```
3. Configure the routing process to serve as an L1/L2 router using the following command:

   ```
   configure isis area area_name is-type level [1 | 2 | both-1-and-2]
   ```

   Specify both-1-and-2 for the level option using the following command:

   Note
   When no other L2 processes are defined on the router, the default IS type level is L1/L2, and this command can be omitted.

4. Add an IS-IS area level-1 address to the router using the following command:

   ```
   configure isis area area_name add area-address area_address
   ```
5. Add an IS-IS area level-2 address to the router using the following command:

   ```
   configure isis area area_name add area-address area_address
   ```
6. Add IS-IS-eligible interfaces to the area using the following command:

   ```
   configure isis add [vlan all | {vlan} vlan_name] area area_name {ipv4 | ipv6}
   ```

   An IS-IS-eligible interface is one that already has the appropriate IP address type (IPv4 or IPv6) address assigned to it.
7. If your topology requires interfaces to operate at a specific topology level, configure the appropriate interfaces with the following command:

   ```
   configure isis [vlan all | {vlan} vlan_name] circuit-type level [1 | 2 | both-1-and-2]
   ```
8. The default IS-IS system ID is the switch MAC address. If you want to change the default IS-IS system ID, use the following command:

   ```
   configure isis area area_name system-id [automatic | system_id]
   ```

   Note
   Although the IS-IS protocols manage IP routing, they use the Connectionless Network Protocol (CLNP) for IS-IS communications between routers. The IS-IS system ID is required to identify the router in the AS.

9. Enable the IS-IS router using the following command:

   ```
   enable isis {area area_name}
   ```

Configuring L2 Routers

To configure a switch to operate as a level 2 IS-IS router, do the following:

1. Prepare the IP interfaces that will support level 2 IS-IS routing as follows:
   a. Add an IPv4 or IPv6 address.
   b. Enable IP or IPv6 forwarding.
2. Create the IS-IS routing process, which is also called an area, using the following command:

   ```
   create isis area area_name
   ```
3. Configure the routing process to serve as a L2-only router using the following command:

   ```
   configure isis area area_name is-type level [1 | 2 | both-1-and-2]
   ```

   Specify 2 for the level option.
4. Add an IS-IS area level-2 address to the router using the following command:

   ```
   configure isis area area_name add area-address area_address
   ```
5. Add IS-IS-eligible interfaces to level 2 using the following command:

   ```
   configure isis add [vlan all | {vlan} vlan_name] area area_name {ipv4 | ipv6}
   ```

   An IS-IS-eligible interface is one that already has the appropriate IP address type (IPv4 or IPv6) address assigned to it.
6. Add an IS-IS system ID to the router using the following command:

   ```
   configure isis area area_name system-id [automatic | system_id]
   ```

   Use the automatic option to assign the switch MAC address to the IS-IS system ID. The default option is **automatic**, so you can also enter the command without options to select the switch MAC address.
7. Enable the ISIS router using the following command:

   ```
   enable isis {area area_name}
   ```

Configuring IS-IS Timers

IS-IS timers allow you to fine tune IS-IS operation. Table 64 lists the IS-IS timers and the command you can use to adjust them.

Table 64: IS-IS Configuration Timers and Commands

Timer	Command
T1 hello restart interval	`configure isis [vlan all | {vlan} vlan_name] timer restart-hello-interval seconds {level [1 | 2]}`
T2 restart timer	`configure isis area area_name timer restart seconds {level [1 | 2]}`
T3 restart grace period	`configure isis restart grace-period seconds`
LSP generation interval	`configure isis area area_name timer lsp-gen-interval seconds {level[1| 2]}`

Table 64: IS-IS Configuration Timers and Commands (continued)

Timer	Command
LSP maximum lifetime	`configure isis area area_name timer max-lsp-lifetime seconds`
LSP refresh interval	`configure isis area area_name timer lsp-refresh-interval seconds`
SPF interval	`configure isis area area_name timer spf-interval seconds {level[1|2]}`
CSNP interval	`configure isis [vlan all | {vlan} vlan_name] timer csnp-interval seconds {level [1 | 2]}`
Hello interval	`configure isis [vlan all | {vlan} vlan_name] timer hello-interval [seconds | minimal] {level [1 | 2]}`
Hello multiplier	`configure isis [vlan all | {vlan} vlan_name] hello-multiplier multiplier {level [1 | 2]}`
LSP retransmit interval	`configure isis [vlan all | {vlan} vlan_name] timer retransmit-interval seconds`
LSP transmit interval	`configure isis [vlan all | {vlan} vlan_name] timer lsp-interval milliseconds`

Configuring the Graceful Restart Feature

The graceful restart feature enables a router process to restart with minimal impact on the network. Without graceful restart, every other IS-IS router in the network must update its LSDB when the router goes down, and this generates a lot of traffic, and it introduces delays while new routes are established. With graceful restart, the router process requests restart support from its neighbors, and the neighbors wait for a predefined time before declaring routes dead. If the restart completes before the predefined time ends, the restart causes minimal impact on the network operation. For more information on graceful restart, see IS-IS Restart.

- To enable or disable the graceful restart feature, use the following commands:

  ```
  enable isis restart-helper
  disable isis restart-helper
  ```

- To configure which events trigger a graceful restart, use the following commands:

  ```
  configure isis restart [ none | planned | unplanned | both ]
  ```

- To configure the timers used to control graceful restart operation, use the commands in Table 65.

Table 65: Timers that Control Graceful Restart

Timer	Configuration Command
T1 restart hello interval	`configure isis [vlan all | {vlan} vlan_name] timer restart-hello-interval seconds {level [1 | 2]}`
T2 restart timer	`configure isis area area_name timer restart seconds {level [1 | 2]}`

Table 65: Timers that Control Graceful Restart (continued)

Timer	Configuration Command
T3 restart grace period	`configure isis restart grace-period seconds`
Hello interval	`configure isis [vlan all | {vlan} vlan_name] timer hello-interval [seconds | minimal] {level [1 | 2]}`
Hello multiplier	`configure isis [vlan all | {vlan} vlan_name] hello-multiplier multiplier {level [1 | 2]}`

Configuring Hello Padding

Hello PDUs are padded to the MTU by default, and this is called hello padding. This is to help identify and correct cases where switches have mismatched MTUs. Because IS-IS packets cannot be fragmented in transit, maintaining consistent MTUs is important to ensure the reliability of packets traversing the network. However, this extra padding could be a waste of bandwidth in scenarios where the MTU is fixed throughout the network.

- To disable hello padding, use the following command:
  `disable isis [vlan all | {vlan} vlan_name] hello-padding`
- To enable hello padding, use the following command:
  `enable isis [vlan all | {vlan} vlan_name] hello-padding`

Configuring Interlevel Filters

Interlevel filters allow you to control how routes are redistributed between IS-IS levels 1 and 2. The redistribution of routes from level 2 to level 1 is called route leaking. For more information, see Route Leaking on page 997.

- To set an interlevel filter for redistribution from level 1 to level 2, use the following command:
  `configure isis area area_name interlevel-filter level 1-to-2 [policy | none] {ipv4 | ipv6}`
- To set an interlevel filter for redistribution from level 2 to level 1, use the following command:
  `configure isis area area_name interlevel-filter level 2-to-1 [policy | block-all | allow-all] {ipv4 | ipv6}`

Configuring the Dynamic Hostname Feature

Dynamic Hostname on page 996 introduces the dynamic hostname feature, which causes text names to replace numbers in some logs.

- To enable the dynamic hostname feature, use the following command:
  `enable isis area area_name dynamic-hostname [area-name | snmp-name]`
- To disable the dynamic hostname feature, use the following command:
  `disable isis area area_name dynamic-hostname`

Configuring the Adjacency Check Feature

When enabled, the adjacency check feature permits an adjacency between two IS-IS interfaces only when both are configured for the same IP address type (IPv4 or IPv6) and are configured for the same subnet.

- To enable the adjacency check feature, use the following command:
  `enable isis area area_name adjacency-check {ipv4 | ipv6}`
- To disable the adjacency check feature, use the following command:
  `disable isis area area_name adjacency-check {ipv4| ipv6}`

Configuring an Import Policy

When applied to a router process, an import policy controls how routes are imported to the FIB from all IS-IS processes on this .

- To apply a policy to a router process, use the following command:
  `configure isis import-policy [policy-map | none]`

Note
The import policy cannot be used to select which routes are added to the routing table. The import policy can only modify the route attributes as routes are added to the routing table.

Configure the Multi-Topology Feature

The multi-topology feature is introduced in IPv4 and IPv6 Topology Modes on page 997.

- To configure the multi topology feature, use the following command:
  `configure isis area area_name topology-mode [single | multi | transition] {level [1 | 2]}`

Displaying IS-IS Information

Displaying General Information for Global IS-IS

- To display general information for global IS-IS, use the following command:
  `show isis`

Displaying Router-Specific Information

- To display router-specific information, use the following command:
  `show isis area [area_name | all]`

Displaying Router Summary Addresses

- To display router summary addresses, use the following command:
  `show isis area area_name summary-addresses`

Displaying IS-IS Interface Information

- To display IS-IS interface information, use the following command:

  ```
  show isis vlan {enabled | { vlan_name | all} }
  ```

Displaying Link State Database Information

- To display link state database information, use the following command:

  ```
  show isis lsdb {area area_name {lsp-id lsp_id}} {level [1|2]} {detail
  | stats}
  ```

Displaying IPv4 and IPv6 Topology Information

- To display IPv4 and IPv6 topology information, use the following command:

  ```
  show isis topology {area area_name {level [1 | 2]}} {ipv4 | ipv6}
  ```

Displaying IS-IS Neighbors

- To display IS-IS neighbor information, use the following command:

  ```
  show isis neighbors {area area_name} {vlan vlan_name} {ipv4 | ipv6}
  {detail}
  ```

Displaying IS-IS Counter Data

- To display IS-IS counter data, use the following command:

  ```
  show isis counters {area [area_name | all] | vlan [vlan_name | all]}
  ```

Managing IS-IS

Configuring Password Security

The ExtremeXOS software supports passwords for the following IS-IS AS components:

- Level 2 domains
- Level 1 areas
- *VLAN* interfaces

Domain and area authentication prevents intruders from injecting invalid routing information into the router. Because passwords must be configured to match at both ends of a connection, password security also helps detect unconfigured and

misconfigured interfaces. After configuration, the password is inserted into LSP, CSNP, and PSNP PDUs and are validated on the receiving end.

Note
The password configuration commands in this section provide an encrypted option, which controls how the passwords are saved and displayed on the switch. The encrypted option does not encrypt messages that are transmitted on the network. All passwords are transmitted in clear text.

- To configure password security for a level 2 domain, use the following command:

  `configure isis area area_name domain-password [none | {encrypted} simple password {authenticate-snp {tx-only}}]`

- To configure password security for a level 1 area, use the following command:

  `configure isis area area_name area-password [none | {encrypted} simple password {authenticate-snp {tx-only}}]`

  Interface authentication prevents unauthorized routers from forming an adjacency. This is achieved by inserting a password in hello PDUs and validating the password on the receiving end. You can configure password protection separately for level 1 and level 2.

- To configure password security for a VLAN interface, use the following command:

  `configure isis [vlan all | {vlan} vlan_name] password [none | {encrypted} simple password] level [1 | 2]`

Managing Transit Traffic with the Overload Bit

If a router lacks the resources to maintain a complete link state database, routing packets to it could result in a blackhole or routing loop. The overload bit prevents this undesirable behavior by indicating to the other routers that the afflicted router should not be used as a transit router. Packets destined for directly attached routes can still be routed to a router with the overload bit set, but all other routing paths reconverge around it.

- To manually enable the overload bit feature, use the following command:

  `enable isis area area_name overload-bit {suppress [external | interlevel | all]}`

- To configure the router to automatically set the overload bit on startup, use the following command:

  `configure isis area area_name overload-bit on-startup [ off | {suppress [external | interlevel | all]} seconds]`

- To disable the overload bit feature, use the following command:

  ```
  disable isis area area_name overload-bit
  ```

  Note

  Enabling or disabling the overload bit feature does not modify the configuration of the command:

  ```
  configure isis area area_name overload-bit on-startup [ off | {suppress [external | interlevel | all]} seconds]
  ```

Clearing the IS-IS Counters

The `show isis counters {area [area_name | all] | vlan [vlan_name | all]}` command can display all counters or only those specific to the router process or the configured VLANs.

- To clear the IS-IS counters for the router and the VLANs, use the following command:

  ```
  clear isis counters
  ```

- To clear the IS-IS counters only for the router process, use the following command:

  ```
  clear isis counters area [area_name | all]
  ```

- To clear the IS-IS counters only for one or all VLANs, use the following command:

  ```
  clear isis counters [vlan all | {vlan} vlan_name]
  ```

Originating an L2 Default Route

This feature injects a zero-cost route to 0.0.0.0 in LSPs originated by an L2 router, thereby advertising the router as the default gateway.

- To enable default route origination, use the following command:

  ```
  enable isis area area_name originate-default {ipv4 | ipv6}
  ```

- To disable default route origination, use the following command:

  ```
  disable isis area area_name originate-default {ipv4 | ipv6}
  ```

Managing IP Summary Addresses

Summary addresses are introduced in Summary Addresses on page 996. L2 routers include in their L2 LSPs a list of all destinations reachable in the L1 area attached to them by default. Summarization of the L1 destinations reduces the amount of information stored on each L2 router and helps in scaling to a large routing domain.

- To add summary addresses to an IS-IS router, use the following command:

  ```
  configure isis area area_name add summary-address [ipv4_address_mask | ipv6_address_mask] {level [1 | 2]}
  ```

- To delete summary addresses from an IS-IS router, use the following command:

  ```
  configure isis area area_name delete summary-address [ipv4_address_mask | ipv6_address_mask] {level [1 | 2]}
  ```

- To display the configured summary addresses for an IS-IS router, use the following command:
  `show isis area area_name summary-addresses`

Managing an IS-IS Area Address

IS-IS was not originally designed for IP routing. Some aspects of CLNP routing, such as system IDs and area addresses, exist in the IP extension of IS-IS. The IS-IS area address is required to identify the area in which the router participates.

- Before you can configure the area address, you must create the IS-IS routing process, which is also called an area, using the following command:
  `create isis area area_name`
- To add an IS-IS area address to the router, use the following command:
  `configure isis area area_name add area-address area_address`
- To delete an IS-IS area address from a router, use the following command:
  `configure isis area area_name delete area-address area_address`

Note
The ExtremeXOS software implementation of IS-IS supports no more than three area addresses.

Managing VLAN Interfaces

Adding a VLAN Interface

- To add a *VLAN* interface to a router, use the following command:
  `configure isis add [vlan all | {vlan} vlan_name] area area_name {ipv4 | ipv6}`

Note
The interface is not enabled separately from the area. If the area is enabled, the interface will begin transmitting hellos as soon as this command is executed, provided the interface is in forwarding mode and has active links.

Setting the VLAN Interface Link Type

The default link type is broadcast. You can change the link type to point-to-point.

- To set the link type, use the following command:
  `configure isis [vlan all | {vlan} vlan_name] link-type [broadcast | point-to-point]`

Setting the VLAN Interface Circuit Type

- The circuit type can be level 1, level 2, or both level 1 and level 2. To set the circuit type, use the following command:
  `configure isis [vlan all | {vlan} vlan_name] circuit-type level [1 | 2 | both-1-and-2]`

Configuring VLAN Interface Metrics

Normally, IS-IS metrics can have values up to 63. These metrics are narrow metrics. IS-IS generates two type, length, and value (TLV) codings, one for an IS-IS adjacency (code, length, and value (TLV) 2) and the second for an IP prefix (TLV 128 and TLV 130). During SPF, if the total cost of the path to a destination exceeds 1023, then according to ISO/IEC 10587, the path is ignored.

To overcome these restrictions, a second pair of TLVs is available, one for IP prefixes (TLV 135) and the second for IS-IS adjacency (TLV 22). With these TLVs, IS-IS metrics can have values up 16,777,215 and the maximum path metric allowed is 4,261,412,864. This allows more flexibility while designing a domain. These metrics are wide metrics.

- To configure the metric style used on the router, use the following command:

  ```
  configure isis area area_name metric-style [[narrow | wide] {transition}] | transition] {level [1 | 2]}
  ```

- To configure the narrow metric used on one or all interfaces, use the following command:

  ```
  configure isis [vlan all | {vlan} vlan_name] metric metric {level[1| 2]}
  ```

- To configure the wide metric used on one or all interfaces, use the following command:

  ```
  configure isis [vlan all | {vlan} vlan_name] wide-metric metric {level[1 | 2]}
  ```

  Note
  Configured narrow and wide metrics for a particular interface must be identical in value while migrating from metric style narrow to metric style wide. Only if the metric style is “narrow” or “wide” (that is, no “transition”) is it okay to have different values (because one of the values is not used).

Configuring the DIS Priority for Broadcast Interfaces

The designated intermediate system (DIS) is introduced in Establishing Adjacencies on page 992. When you configure the DIS priority, higher priority means higher election precedence (60 gets elected before 50).

- To configure the DIS priority, use the following command:

  ```
  configure isis [vlan all | {vlan} vlan_name] priority priority {level[1 | 2]}
  ```

  Note
  DIS priority 0 is the lowest priority value. Unlike an *OSPF* DR election, a DIS priority of 0 does not make a router ineligible for DIS election.

Configuring Interface Participation in a Mesh Environment

Fully-meshed point-to-point topologies suffer from a potentially large amount of needless LSP flooding. At the trade-off of resiliency, interfaces can be restricted from flooding received LSPs.

- To configure an interface for a mesh environment, use the following command:

  `configure isis [vlan all | {vlan} vlan_name] mesh [block-none | block-all | block-group group_id]`

  a. To block the flooding of received LSPs altogether, use the mesh block-all option.
  b. To block the flooding of LSPs received on a specific group of interfaces (which are those with the same mesh group ID), use the mesh block-group *group_id* option.
  c. To remove blocking for an interface, use the mesh block-none option.

Resetting a VLAN Interface to the Default Values

- To reset a *VLAN* interface to the default values, use the following command:

  `unconfigure isis [vlan all | {vlan} vlan_name] {level [1|2]}`

Deleting a VLAN Interface

- To delete a *VLAN* interface from a router, use the following command:

  `configure isis delete [vlan all | {vlan} vlan_name] {area area_name} {ipv4 | ipv6}`

Managing IS-IS Routers

Adding an IS-IS Router

- To add an IS-IS router, use the following command:

  `create isis area area_name`

Changing the IS-IS Level of a Router

- To change the IS-IS level of a router, use the following command:

  `configure isis area area_name is-type level [1 | 2 | both-1-and-2]`

Resetting an IS-IS Router to the Default Values

- To reset an IS-IS router to the default values, use the following command:

  `unconfigure isis area area_name {level [1|2]}`

Restarting All IS-IS Routers in a Virtual Router

- To restart all IS-IS routers in a virtual router, use the following command and specify the IS-IS process:

  `restart process [class cname | name {msm slot}]`

  For example:

  ```
  restart process isis
  ```

Disabling an IS-IS Router

- To disable an IS-IS router, use the following command:

  `disable isis {`**`area`** *`area_name`*`}`

Deleting an IS-IS Router

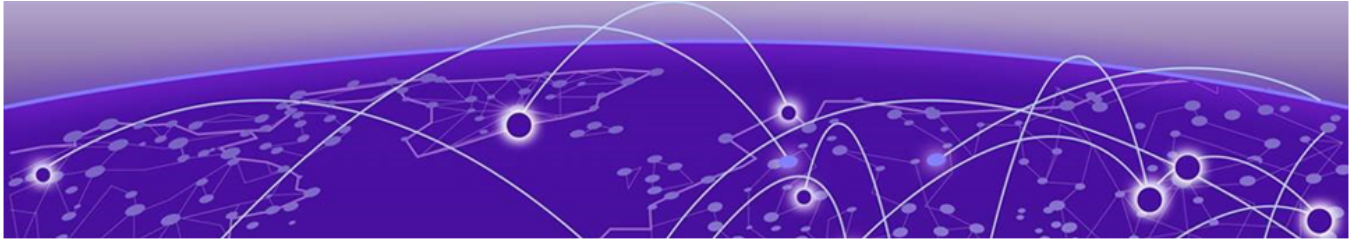
- To delete an IS-IS router, use the following command:

  `delete isis area [`**`all`** `|` *`area_name`*`]`

Configuration Example

Below is a sample configuration of CFM support in *EAPS (Extreme Automatic Protection Switching)*:

```
switch 1 # sh configuration cfm
#
# Module dot1ag configuration.
#
create cfm domain string "MD1" md-level 6
configure cfm domain "MD1" add association string "MD1v1" vlan "v1"
configure cfm domain "MD1" add association string "MD1v2" vlan "v2"
configure cfm domain "MD1" association "MD1v1" ports 17 add end-point down 6
configure cfm domain "MD1" association "MD1v1" ports 23 add end-point down 5
configure cfm domain "MD1" association "MD1v2" ports 31 add end-point down 13
configure cfm domain "MD1" association "MD1v1" ports 17 end-point down add group
"eapsCfmGrp1"
configure cfm domain "MD1" association "MD1v1" ports 23 end-point down add group
"eapsCfmGrp2"
configure cfm domain "MD1" association "MD1v2" ports 31 end-point down add group
"eapsCfmGrp3"
configure cfm group "eapsCfmGrp1" add rmep 2
configure cfm group "eapsCfmGrp2" add rmep 4
configure cfm group "eapsCfmGrp3" add rmep 12
switch 2 # sh configuration "eaps"s
#
# Module eaps configuration.
#
enable eaps
create eaps d1
configure eaps d1 mode transit
configure eaps d1 primary port 17
configure eaps d1 secondary port 23
enable eaps d1
create eaps d2
configure eaps d2 mode transit
configure eaps d2 primary port 31
configure eaps d2 secondary port 23
enable eaps d2
configure eaps d1 add control vlan v1
configure eaps d1 add protected vlan pv1
configure eaps d2 add control vlan v2
configure eaps d2 add protected vlan pv2
create eaps shared-port 23
configure eaps shared-port 23 mode partner
configure eaps shared-port 23 link-id 100
configure eaps cfm add group eapsCfmGrp1
configure eaps cfm add group eapsCfmGrp2
configure eaps cfm add group eapsCfmGrp3
```



Keychain Manager

Keychain Manager Overview on page 1013
Configuring and Using Keychain Manager on page 1015

This chapter introduces Keychain Manager (KCM), a convenient and centralized way for the user to store and manage keys in ExtremeXOS. KCM allows the user configure sets of keys in keychains with fixed periods of validity. It manages the selection, activation, and expiry of keys for the keychains, and it handles rollover to new keys when active keys expire.

Keychain Manager Overview

Keychain Manager (KCM) creates and manages authentication keys in ExtremeXOS. In KCM, all keys are grouped into sets called keychains. KCM stores keychains and manages the activation, expiration, and rollover of keys.

When an ExtremeXOS application registers to use a keychain, KCM informs the application of key-related events and provides information about keys the application needs for authentication.

Keys and Keychains

A keychain contains up to 8 keys. Each key has a *key identifier*, or key ID, that is unique within the keychain, and a secret *key string* that is used for authentication of protocol packets.

Each key has a cryptographic algorithm, which is used with the key string to calculate the key's hash value. You select an algorithm for each key: either HMAC-SHA-1, HMACSHA-256 (the default), HMAC-SHA-384, or HMAC-SHA-512. All of the algorithms are NIST FIPS 180-4 compliant. Keys in the same keychain can have different algorithms.

Key Lifetimes

An *active key* is the key currently being used by the applications registered to a keychain. Within a keychain, only one key can be active at a time. When an active key expires, KCM attempts to roll over to a new active key. KCS selects the new active key based on *key lifetimes* you have defined.

You configure a key's lifetime -- the time period during which it is valid -- by specifying a start time (a date and time, either in UTC or local time) and an end time (the duration,

in seconds, from the start time). The maximum lifetime for a key is 180 days. You need to define only one lifetime for each key, because ExtremeXOS uses the same value for a key's send lifetime, used for outgoing communication, and its accept lifetime, used for incoming communication.

There is no default lifetime. As a result, a key cannot become active until you configure its lifetime.

How KCM Manages Keys and Keychains

When an application is registered to use a keychain, and the active key expires, KCM selects a new active key from within the keychain. The new active key is the one with the earliest start time, unless its end time is in the past. If more than one key has the earliest start time, the key with the latest end time is selected. If there is still no clear choice, then the key with the lowest key ID is selected. (If all of the keys' end times are in the past, then no key is selected.)

Note
Time changes, for example moving to daylight saving time, can affect KCM operations. When a time change occurs, active keys are cancelled and KCM attempts to assign new active keys. However, when a time zone is changed on a switch, KCM operation is not affected. Key lifetimes configured in local time are changed to reflect the new time zone.

When an active key expires and when a new key becomes active, KCM notifies the registered applications for that keychain. The notification includes the key string for the new active key.

We recommend that you configure keys in each keychain so that keys roll over at predetermined times. However, you can configure a grace period of up to 600 seconds so that a recently expired key can be accepted for incoming packets by applications that support the feature.

You can delete a key unless it is the active key for a keychain. However, you cannot change a key's key string.

Supported Platforms

All ExtremeSwitching Universal switches.

Note
OSPFv3 is the only application that currently uses Keychain Manager.

Keychain Manager CLI Commands

`configure keychain` *`keychain_name`* **`accept-tolerance`** *`seconds`*

```
configure keychain keychain_name add key key_id key-string [text_string
active-lifetime local start start_time [end end_time | [duration
[seconds | maximum]]] | encrypted encrypted_string]

configure keychain keychain_name key key_id active-lifetime local start
start_time [end end_time | [duration [seconds | maximum]

configure keychain keychain_name key key_id hash-algorithm algorithm

create keychain keychain_name

delete keychain keychain_name

show keychain keychain_name detail
```

Configuring and Using Keychain Manager

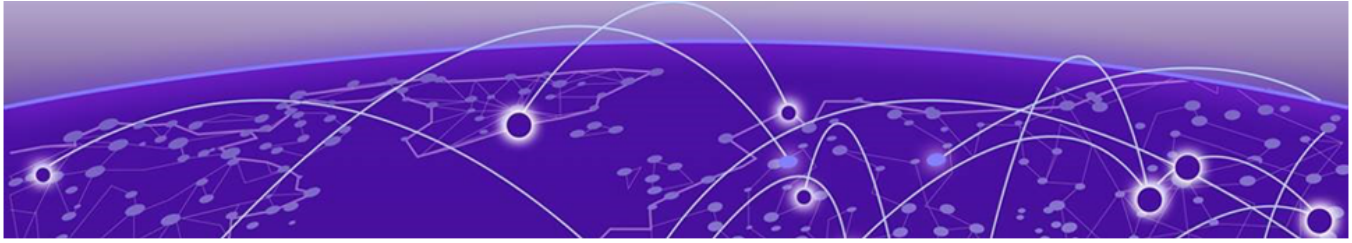
Extreme recommends to synchronize the clocks on all devices that will use Keychain Manager.

Complete the following steps in order to configure Keychain Manager:

1. Create a keychain using the following command:

   create keychain keychain_name
2. Add keys to the keychain and configure options with the following commands:

   ```
   configure keychain keychain_name add key key_id key-string
   [text_string {active-lifetime {local} start start_time [end
   end_time | [duration [seconds | maximum]]]} | encrypted
   encrypted_string]configure keychain keychain_name key key_id active-
   lifetime {local} start start_time [end end_time | [duration [seconds
   | maximum]]]configure keychain keychain_name key key_id hash-algorithm
   algorithm

   configure keychain keychain_name key key_id active-lifetime
   {local} start start_time [end end_time | [duration [seconds |
   maximum]]]configure keychain keychain_name key key_id hash-algorithm
   algorithm

   configure keychain keychain_name key key_id hash-algorithm algorithm
   ```
3. Configure a supported client to use keychain. The authentication peer of the client must use the same key information.



Layer 2 Tunneling and Filtering

Layer 2 Protocol Tunneling on page 1016
Protocol Tunneling on page 1018
Protocol Filtering on page 1021
L2PT Limitations on page 1023

This ExtremeXOS feature introduces the ability to tunnel and filter Layer 2 PDUs. Tunneling allows you to send Layer 2 PDUs across a service provider network, and be delivered to remote switches. It is useful when a network includes remote sites that are connected through a service provider network. Using tunneling, you can make the service provider network transparent to the customer network.

Filtering prevents Layer 2 PDUs from being received on a port.

Layer 2 Protocol Tunneling

Layer 2 protocol tunneling (L2PT) is achieved by encapsulating the PDUs at the ingress PE device before transmitting them over the service provider network. The encapsulation prevents the PDUs from being processed by the switches in the SP network. At the egress PE device, the encapsulated packets are de-encapsulated, and transmitted to the CE device.

The encapsulation used for different types of networks is as follows:

- *VLAN (Virtual LAN)*/VMAN – The Destination Address (DA) MAC of the Layer 2 PDU is changed to the L2PT DA MAC. The switch shall also add any VLAN tags that may be required to the Layer 2 PDU before transmitting over the SP network.
- VPLS/VPWS – The DA MAC of the Layer 2 PDU is changed to L2PT DA MAC. The Layer 2 PDU is then treated like any other data packet by the *MPLS (Multiprotocol Label Switching)* stack. The MPLS stack shall add the labels and L2 headers as per its configuration to the Layer 2 PDU before transmitting over the SP network.
- VXLAN – The DA MAC of the Layer 2 PDU is changed to L2PT DA MAC at the ingress remote tunnel end-point (RTEP). The modified packet is then encapsulated into a VXLAN packet and sent over the network. At the egress RTEP, the packet is lifted to the CPU for L2PT processing. After VXLAN decapsulation, the DA MAC is changed from L2PT MAC to the protocol MAC and is sent on the access ports of the tenant VLAN.

Tunneling is configured on a service by specifying a tunneling action for each interface of the service. The possible actions are:

- Tunnel – Configuring an interface of a service to tunnel for a protocol enables the interface to tunnel PDUs of the configured protocol that are received by the underlying port of the interface. Any PDUs that are received in its native format are tunneled instead of processing locally by the switch. Any PDUs of the protocol that are received in its encapsulated format are dropped by the switch (receiving an encapsulated packet on an interface configured to tunnel is considered proof of network misconfiguration, or loops).
- Encapsulate/Decapsulate – Configuring an interface of a service to encapsulate or de-encapsulate for a protocol enables the interface to transmit and receive PDUs of that protocol in its encapsulated format. Native PDUs of the protocol may still be received by the underlying port of the interface, but they will not be tunneled and instead are processed locally by the switch.
- None – Configuring an interface of a service to none for protocol marks the interface as not participating in tunneling for that protocol. Native PDUs of the protocol that are received on the underlying port of the interface shall either be processed locally by the switch or be tunneled by another service which is configured to tunnel that protocol. Encapsulated PDUs that are received on the interface are treated like any other L2 packet.

An operator can specify a *CoS (Class of Service)* value for the tunneled PDUs. This can be useful since some L2 protocols may have a higher priority than others (for example, *STP (Spanning Tree Protocol)* may be considered higher priority than *LLDP (Link Layer Discovery Protocol)*). If a CoS value is specified for a protocol for which tunneling is enabled, the switch will transmit the encapsulated PDUs for that protocol with the operator specified CoS towards the network. The CoS value specified by the operator is transmitted on the SP network as follows:

- VLAN/VMAN – The CoS value is written to the PRI bits of the outermost VLAN tag if available.
- VPLS/VPWS – The CoS value is written to the EXP bits of the outermost MPLS label. The action taken by the switch for PDUs of a protocol is as described in the following table.
- VXLAN – The CoS value configured on the profile attached to the access port is written to the PRI bits of the outer VLAN header of the VXLAN encapsulated frames before transmitting them to other RTEPs.

As VXLAN tunneled packets cross L3 boundaries in the underlay network, the CoS can get lost when traversing L3 boundaries. An operator may choose to configure a Differentiated Services Code Point (DSCP) that needs to be set in the outer IP header of the encapsulated packets. If the packet encapsulated into the VXLAN tunnel is an IP packet, the DSCP from inner IP header is typically copied to DSCP of the outer IP header. A configuration option is provided to overwrite this outer DSCP value. In case of

L2 protocols (which do not have an inner DSCP), the configured DSCP value is set in the outer IP header.

Table 66: L2 PDU Actions

Ingress Action	Egress Action	Switch Action
None or Encap/Decap	NA	Process locally
Tunnel	None	Discard PDU at egress
Tunnel	Tunnel	Tx PDU natively
Tunnel	Encap/Decap	Tx PDU encapsulated

The action taken by the switch for encapsulated PDUs for a protocol is as described in the following table.

Table 67: L2 Encapsulated PDU Actions

Service has at least one I/F with tunnel action	Ingress Action	Egress Action	Switch Action
No	None or Encap/ Decap	None or Encap/ Decap	Forward
Yes	None or Tunnel	NA	Discard packet at ingress
Yes	Encap/Decap	None	Discard packet at egress
Yes	Encap/Decap	Tunnel	Tx PDU natively
Yes	Encap/Decap	Encap/Decap	Tx PDU encapsulated

Protocol Tunneling

To make L2PT configuration easier, in ExtremeXOS you can create L2PT profiles. An L2PT profile specifies the tunneling action and other parameters for protocols (specified using protocol filters) that should be tunneled. You can then apply the profile to the interfaces of the service that are participating in L2PT. And you can also change the profile when it is already bound to an interface.

The L2PT parameters that can be configured through a profile include the following:

- Tunneling Action
- Tunneling CoS
- Tunneling DSCP for VXLAN case

The following validity checks are performed when an entry for a protocol filter is created in an L2PT profile:

- Ensure that all protocols in the protocol filter define a destination MAC address.
- Ensure that all protocols in the protocol filter define a protocol identifier.
- Ensure that all protocols in the protocol filter are unique within the L2PT profile.
- If the action for the protocol filter is ‘encapsulate:
  - Ensure that there are no entries with action as ‘tunnel in the L2PT profile.

    - Ensure that the service interface is either a tagged *VLAN* port, a PW, or a VXLAN RTEP.

- If the action for the protocol filter is 'tunnel':
    - Ensure that there are no entries with action as 'encapsulate' in the L2PT profile.
    - For every service interface using the L2PT profile:
        - Ensure that none of the protocols in the protocol filter are filtered on the underlying port of the interface.
        - Ensure that none of the protocols in the protocol filter are tunneled on the underlying port of the interface.

The following validity checks are performed when a L2PT profile is bound to an interface of a service:

- If the profile specifies the action as 'tunnel' for protocol filter:
    - Ensure that the interface is not a PW or a VXLAN RTEP.
    - Ensure that none of the protocols in the L2PT profile are filtered on the underlying port of the interface.
    - Ensure that none of the protocols in the L2PT profile are tunneled on the underlying port of the interface.

Typically, you will want to configure the tunneling action for all customer facing interfaces of the service that participate in L2PT as tunnel, and the tunneling action for all network facing interfaces as encapsulate/decapsulate. Once any interface of the service is configured to tunnel a protocol, the switch will configure all tagged ports and PWs or VXLAN RTEPs of the service to encapsulate/decapsulate mode. You can override this implicit configuration by binding the RTEPs a profile to the service interface that specifies a different tunneling action.

For example, consider a VMAN service named c1 with customer facing ports 1, 2 and 3 and network facing ports 4, 5, 6. Ports 4, 5 and 6 are added as tagged to the VMAN and 1, 2 and 3 are added as untagged to the VMAN. The operator wants to tunnel LACP and EFM OAM on all customer facing ports at CoS 5. The configurations that he or she must make are as follows:

```
# Create a protocol filter
create protocol filter "my_slow_protocols_filter"

# Add LACP to the protocol filter
configure protocol filter "my_slow_protocols_filter"
add dest-mac 01:80:C2:00:00:02 etype 0x8809 field offset 14 value 01 mask FF

# Add EFM OAM to the protocol filter
configure protocol filter "my_slow_protocols_filter"
add dest-mac 01:80:C2:00:00:02 etype 0x8809 field offset 14 value 03 mask FF

# Create an L2PT profile for the customer facing ports named c1_l2pt_profile
create l2pt profile "c1_l2pt_profile"

# Enable CDP tunneling with CoS 5
configure l2pt profile "c1_l2pt_profile" add protocol filter
"my_slow_protocols_filter" action tunnel cos 5

# Enable CDP tunneling with CoS 3 and DSCP 38 in case of VXLAN tenant VLAN
configure l2pt profile "c1_l2pt_profile" add protocol filter
```

```
"my_slow_protocols_filter" action tunnel cos 5 dscp 38

# Bind c1_l2pt_profile to all customer facing ports
configure vman c1 ports 1,2,3 l2pt profile "c1_l2pt_profile"

# Please note that the network facing port 4, 5 and 6 don't have to be explicitly
# configured to encapsulate/decapsulate mode since the switch implicitly sets all
# tagged ports to encapsulate/decapsulate mode when an L2PT profile is bound to
# any port of the service.
```

The operator also has the option to configure the L2PT destination MAC address (i.e., the DA used by L2PT encapsulated PDUs). This is may be done using the following CLI command:

```
configure l2pt encapsulation dest-mac mac_address
```

The L2PT destination MAC address may only be changed when no L2PT profiles have been bound to any service interface. The default L2PT DA MAC is 01:00:0C:CD:CD:D0 (selected to be interoperable with Cisco and Juniper).

Use the following commands to view the status and statistics of L2PT:

```
show [vlan | vman] vlan_name {ports port_list} l2pt {detail}

show {l2vpn} [vpls vpls_name | vpws vpws_name] {peer ipaddress} l2pt
{detail}

show [[{vlan} vlan_name] {{vxlan {vr vr_name} rtep rtep_ipv4}}] l2pt
{detail}
```

Use the following commands to clear L2PT stats:

```
clear l2pt counters {[vlan | vman] vlan_name {ports port_list}}

clear l2pt counters {[vpls vpls_name {peer ipaddress} | vpws vpws_name]}

clear l2pt counters {[vlan] vlan_name {{vxlan {vr vr_name} rtep rtep_ipvs}}}
```

Implementing L2PT in ExtremeXOS

In ExtremeXOS, the L2PT data-plane is implemented almost entirely in software. When you attach a L2PT profile to a service interface, the following *ACL (Access Control List)* rules are configured:

- An ACL rule is added to copy and drop all packets with a destination address equal to the L2PT destination MAC address, and an outer *VLAN* ID equal to the VLAN tag of the service. In case of VXLAN tenant VLAN, the condition for VLAN ID is modified so that the packet is copied to CPU on any VLAN received. The traffic for the destined tenant VLAN ID can be received on any underlay L3 VLAN.
- For each protocol that is tunneled on the service interface, an ACL rule is added to copy and drop all packets with the same the destination address as the protocol. If the protocol defines an EtherType, then the rule is also qualified with the EtherType.

- If any protocol is tunneled on the service interface, an ACL rule is added to drop all packets received on the service interface with a destination address equal to the L2PT destination MAC address.

Protocol Filtering

You can enable filtering of PDUs of a protocol on any port. If you enable filtering for a protocol on a port, the switch discards PDUs of that protocol on that port.

Use the following command to view protocol filter status and statistics:

```
show ports [port_list | all] protocol filter {detail}
```

Use the following command to clear protocol filtering stats:

```
clear counters ports {port_list} protocol filter
```

Implementing Protocol Filtering in ExtremeXOS

In ExtremeXOS, the protocol filtering data-plane is implemented partially in hardware and partially in software. Filtering is performed only on the ingress. When a protocol filter is attached to a port, the following *ACL* rules are configured:

- For each protocol in the protocol filter: If the protocol does not define a user-defined field, and the protocol identifier is EtherType, or does not have a protocol identifier:
  - An ACL rule is added to drop all packets on the port that match the destination address of the packet. The rule is also qualified with the EtherType of the protocol if it defines one.

  Else:
  - An ACL rule is added to copy and drop all packets on the port that match the destination address of the packet. The rule is also qualified with the EtherType of the protocol if it defines one.

The protocol filtering data-plane inspects all packets received from ports that have protocol filters attached, and drops any packet that matches any of the protocols configured in the protocol filter.

Protocol filtering is also supported for VXLAN tenant VLANs. However, filtering has to be configured on access ports only, and is not supported on RTEPs. This aligns with the existing behavior where filtering cannot be supported on pseudo wires.

Protocol Filters

Both L2PT and protocol filtering allow you to tunnel or filter many protocols on an interface. For this purpose, ExtremeXOS supports creating protocol filters. A protocol filter contains a number of protocols to which you can apply some action (like tunneling and filtering). Each protocol in a protocol filter is defined using the following fields:

- The destination MAC address of PDUs of the protocol. This field is mandatory for all protocols that are to be tunneled or filtered.

- The protocol id (EtherType, LLC, SNAP). This field is mandatory for all protocols that are to be tunneled.
- User defined field. This is an arbitrary field in the PDU of the protocol that is specified using the offset of the field from the start of the PDU, the value of the field and a mask.

For example, use the following command to create a protocol filter that includes LACP and EFM OAM:

```
# Create a protocol filter
create protocol filter my_slow_protocols_filter

# Add LACP to the protocol filter
configure protocol filter my_slow_protocols_filteradd dest-mac
01:80:C2:00:00:02 etype 0x8809 field offset 14 value 01 mask FF

# Add EFM OAM to the protocol filter
configure protocol filter my_slow_protocols_filteradd dest-mac
01:80:C2:00:00:02 etype 0x8809 field offset 14 value 03 mask FF
```

The following validity checks are performed when a protocol is added to a protocol filter:

- Ensure that the protocol does not already exist in the protocol filter.
- If the protocol filter is used by any L2PT profile:
  - Ensure that the protocol defines a destination MAC address.
  - Ensure that the protocol defines a protocol identifier.
- For every L2PT profile that is using the protocol filter:
  - Ensure that the protocol is unique within the L2PT profile. If the action for the protocol filter is 'tunnel' in the L2PT profile:
    - For every service interface using the L2PT profile: ensure that the protocol is not filtered on the underlying port of the service interface.
    - It ensures that the protocol is not tunneled on the underlying port of the service interface.
- If the protocol filter is used by any port for the purpose of protocol filtering (`configure ports port# protocol filter filter-name`):
  - Ensure that the protocol defines a destination MAC address.
- For every port that has the protocol filter attached for the purpose of protocol filtering:
  - Ensure that the protocol is not tunneled by a service on that port.

Note

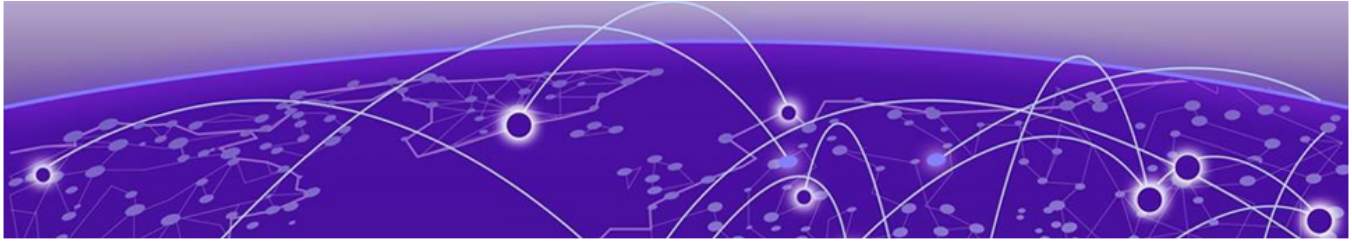
Protocol filters may be used with features other than L2PT and protocol filtering (for example, Protocol Based VLANs). The validity tests listed above are only the ones relevant to L2PT and protocol filtering.

Protocol filters for the following protocols are created automatically by the switch when the switch is set to default configuration:

- Cisco Discovery Protocol (CDP)
- Unidirectional Link Detection (UDLD)
- *VLAN* Trunking Protocol (VTP)
- Port Aggregation Protocol (PAgP)
- Dynamic Trunking Protocol (DTP)
- Link Aggregation Control Protocol (LACP)
- *LLDP*
- *STP*
- *EDP (Extreme Discovery Protocol)*

L2PT Limitations

- L2PT and protocol filtering is implemented in software, so the number of frames that can be filtered or tunneled is limited.
- Both L2PT and protocol filtering can be configured only through CLI. Configuration through *SNMP (Simple Network Management Protocol)*/XML is not supported for this release.
- If L2PT configurations are made on PWs, these configurations are lost on a restart of the *MPLS* process unless the L2PT process is also restarted.
- If L2PT configurations are made on a VPLS or VPWS service, dot1p tag inclusion must be enabled on the VPLS/VPWS. Also, VPLS or VPWS service must have same *VLAN* IDs at each end of a PW.
- When tunneling protocols are point-to-point in nature, it is your responsibility to ensure that there are only two tunnel endpoints for the protocol.
- If a protocol that is configured to be tunneled on a service interface cannot be uniquely identified by its destination address and EtherType, then all packets with the same DA and EtherType of the protocol being tunneled (but that are not really PDUs of the protocol) will be slow path forwarded.
- Tagged protocol PDUs cannot be tunneled over VLANs. Tagged protocol PDUs can only be tunneled over VMANs (the VMAN can be the service VMAN for a VPLS/VPWS service, or a standalone VMAN). Untagged protocol PDUs can be tunneled over both VLANs and VMANs (the VLAN/VMAN can be standalone, or be the service VMAN for a VPLS/VPWS service).
- Untagged protocol PDUs cannot be bypassed if the ingress port is an untagged VMAN port with a default CVID. Untagged protocol PDUs can be bypassed if the ingress port is an untagged VMAN port without a default CVID.
- In VPLS, only full-mesh configuration is supported for L2PT.
- L2PT is not supported on VLAN ports that have a port specific tag.
- L2PT is not supported on provider edge ports when MACSEC was enabled on customer edge.”



Layer 3 Virtual Private Network

Overview of Layer 3 VPN on page 1024
Overview of BGP/MPLS Network on page 1025
Overlapping Customer Address Spaces on page 1028
Multi-protocol BGP Extension on page 1028
Multiple Forwarding Tables on page 1029
Quality of Service in BGP/MPLS VPN on page 1029
Virtual Routing and Forwarding Instances on page 1029
L3VPN BOOTP Relay on page 1030
L3VPN Configuration Example on page 1030

This chapter introduces Layer 3 VPN, a way to create a tunnel between customer sites through a Provider Backbone, and its features and options in a *BGP (Border Gateway Protocol)*/*MPLS (Multiprotocol Label Switching)* VPN environment. This chapter also provides an example showing L3VPN configuration on a BGP/MPLS setup.

Overview of Layer 3 VPN

Layer 3 Virtual Private Networks (L3VPN) is a specific implementation of PPVPN (Provider Provisioned VPN). L3VPN is a way to create a tunnel between customer sites through a Provider Backbone, and that tunnel is established and maintained by the Service Provider.

Within a Layer 3 VPN, the customer advertises IP routing knowledge to the provider network. The ISP then advertises this routing information across its network to the other customer locations. This simple concept requires some coordination between the provider and the customer. Also, the provider must configure its network to support the advertisement of these routes and have a way to segregate the customer routes. This is accomplished with the help of a specially designed Network Layer Reachability Information (NLRI).

Within a L3VPN, several routers will play a different role. On the customer site, there is a CE router (Customer Edge). This router is the property of the customer and is managed by them. This router is outside of the Autonomous System (AS) of the Provider. On the Provider side there are some PE routers (Provider Edge) and P routers (Provider). The PE routers are facing the CE routers. They have all the customer's IP routing knowledge. The P routers do not have that knowledge; they are not facing any CE routers, and they serve only to transport the data from PE routers to other PE routers. Their knowledge is very limited, and they usually are intended only for swapping *MPLS* labels. In other

words the Provider's core is pure MPLS and has no knowledge of L3VPN, while the edge is L3VPN aware.

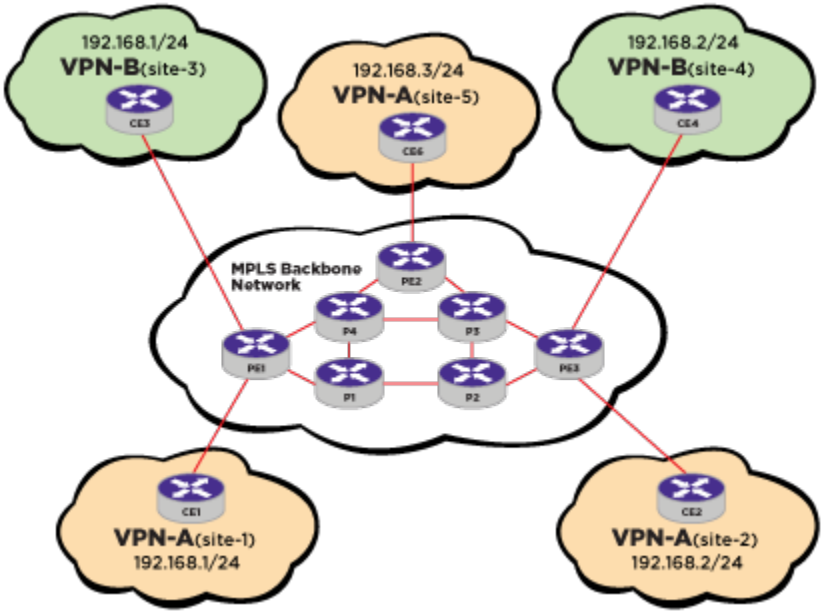


Figure 114: Different Router Roles in L3 VPN

A L3VPN requires an MPLS transport mechanism in the Provider Backbone for the data forwarding and Multiprotocol *BGP* (MBGP) between the PE routers to exchange VPN-IPv4 routing information.

Overview of BGP/MPLS Network

This section introduces the basic concepts of *BGP*/*MPLS* VPNs as described in the Internet RFC 4364. Figure 115 illustrates an example BGP/MPLS VPN network configuration that includes two VPNs: VPN-A comprises three sites, and VPN-B comprises two. Both VPNs use the same set of IP addresses as defined in RFC 1918.

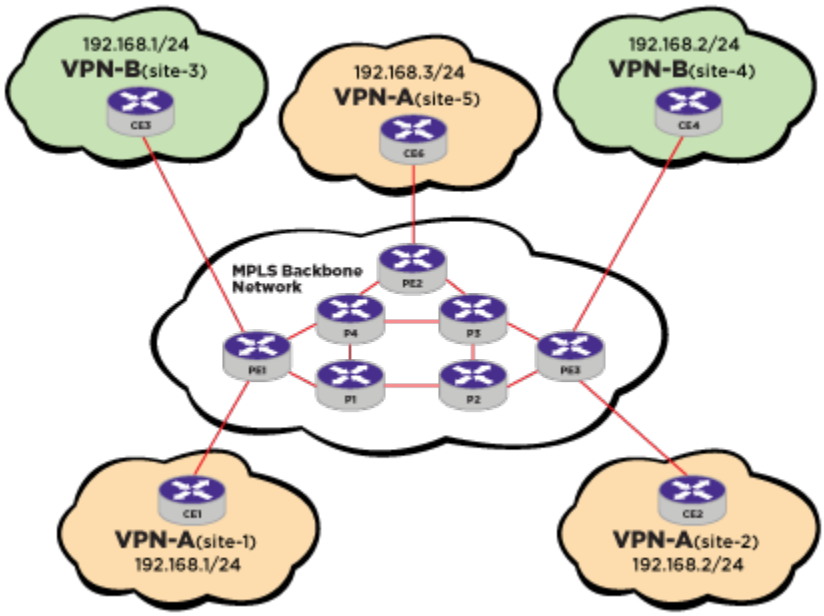


Figure 115: BGP/MPLS VPN Example

The Customer Edge boxes represent switches or routers, while the Provider Edge and P(rovider) boxes are routers. If the CE box is a router, then it only needs to peer with the adjacent PE router(s). The PE and P routers form the core MPLS network. The P routers only maintain routing information for the core MPLS network, while the PE routers only maintain routing information for the VPNs they support. This ensures that no single router maintains routing information for all the VPNs.

Since a PE router can support multiple VPNs that may have overlapping address spaces, each PE router must maintain multiple Virtual Routing and Forwarding tables (VRFs). In the example, each site is a member of one VPN, and the PE router is configured to associate a particular VPN with each physical interface to a CE router. The PE router maintains one VRF for each VPN, and packets received from a particular physical interface are forwarded using the VRF for the VPN associated with that interface.

VRFs for a specific VPN are populated in two ways: (1) when the PE router learns routes from a directly-attached CE router that is a member of the VPN, and (2) when the PE router receives routes for the VPN through MBGP from another PE router. The PE router can learn the routes that are reachable at a particular CE router's site through configuration (static routes), or by routing protocol exchanges with the CE router.

VPN route distribution uses BGP-4 Multiprotocol Extensions that enable BGP to carry routes from multiple address families. The VPN-IPv4 address family supports BGP/MPLS VPNs. A VPN-IPv4 address is a 12-byte entity, beginning with an 8-byte (RD) and ending with a 4-byte IPv4 address. The RD makes it possible to create distinct routes to a common IPv4 address prefix, which is necessary when the same IPv4 address prefix is used in two different VPNs.

The purpose of the RD is solely to allow one to create distinct routes to a common IPv4 address prefix. The Route target attribute is used to identify a set of VRFs. Every VRF is

associated with one or more route targets. Export route targets identify the set of route targets that a PE router attaches to a route learned from a particular CE site. Import route targets identify the set of route targets that determine whether a route received from another PE router can be inserted in a particular VRF. A VPN-IPv4 route can only be installed in a particular VRF if there is a route target that is both one of the route's targets, and one of the VRF's import route targets. Route targets allow a PE router to only maintain routing information for the VPNs it is supporting. Using import and export route targets offers flexibility in constructing a variety of VPN topologies (such as a fully-meshed closed user group, or a hub-and-spoke architecture). Route Targets are encoded as BGP Extended Communities attributes.

When distributing a VPN route through BGP, a PE router includes its own IP address as the BGP next hop (next-hop-self). The PE router always assigns and distributes an MPLS label for each customer VPN VRF that it receives from directly connected sites (CEs). BGP-distributed MPLS labels require that there be a label switched path between the PE router that installs the BGP-distributed route and the PE router that is the BGP next hop of that route. This is necessary because a multi-label stack is used to forward VPN packets across the MPLS backbone.

The outer MPLS label gets the packet across the backbone. This label is obtained from the MPLS signaling protocols, and is associated with the best LSP to the BGP next hop address of the PE Router that advertised the VPN route. The inner MPLS label is obtained from BGP, and is associated with the best route to the VPN destination address. This label identifies the VRF that the egress PE Router uses to forward the packet to a CE device, and may indicate the outgoing interface that the packet should be forwarded over (along with the appropriate link layer header for that interface).

The use of a two-level MPLS label stack is an important scalability attribute of the model, because it is the two-level label stack that enables the P routers to operate without containing routes for any of the VPNs.

In summary, key aspects of the BGP/MPLS VPN model include:

- Direct peering of customer routers with service provider routers.
- Maintenance of multiple forwarding tables by PE routers.
- Introduction of the VPN-IPv4 address family.
- Constrained distribution of routing information via Route Targets.
- Use of MPLS label switching in the backbone network.

VPNV4 NLRI

One of the core principles of operating a VPN is maintaining separation between the customer networks. L3VPN uses a special format to represent customer routes within the provider's network. This format allows each provider router to uniquely view routes from different customers as separate, even when they advertise the same IPv4 prefix. The format consists of these fields: Mask, MPLS label, route distinguisher, and IPv4 prefix.

Limitations

The following list identifies the limitations of the L3VPN feature:

- *RIP (Routing Information Protocol)* is not supported for PE – CE peering routing protocol.
- IP Multicast BGP/MPLS VPN is not supported
- OSPFv2 and ISIS is not supported for PE-CE peering routing protocol.
- IPv6 VPN is not supported.
- Graceful restart mechanism for BGP with MPLS (RFC-4781) is not supported.
- Constraint Route distribution for BGP/MPLS VPN (RFC-4684) is not supported.
- Carrier of carriers BGP/MPLS VPN configuration is not supported.
- XML support to configure BGP/MPLS VPN parameters.
- Carrier's carrier (RFC 4364, Section 9).
- Inter-AS / inter-provider VPNs (RFC 4364, Section 10).
- No other BGP related MIB other than RFC 1657 is supported.

Overlapping Customer Address Spaces

VPN customers often manage their own networks and use the RFC-1918 private address space. If globally unique IPv4 addresses are not used, the same 32-bit IPv4 address can be used to identify different systems in different VPNs. This causes routing problems because *BGP* assumes that each IPv4 address that it carries is globally unique. To solve this problem, BGP/*MPLS* VPN converts non-unique IPv4 addresses into unique VPN-IPv4 address families. Each IPv4 address is prepended with an 8-bytes long Route Distinguisher (RD).

A VPN-IPv4 address is a 12-byte quantity composed of an 8-byte RD and 4 byte IPv4 address. The 8-byte RD is composed of a 2-bytes type field and a 6-bytes value field. The value of the Type field determines the lengths of the Value field's two subfields (Administrator and Assigned Number), as well as the semantics of the Administrator field.

Most commonly, the Admin field holds a 2-byte non-private AS number and the Assigned number field holds a 4-byte number assigned by the VPN service provider. Another common format for the RD is the Admin field holds a global 4-byte IPv4 address, and the Assigned number field holds 2 byte number assigned by the VPN service provider.

Multi-protocol BGP Extension

BGP is originally designed to carry routing information for IPv4 address families only. Since VPN-IPv4 addresses are 12 bytes wide, BGP requires some kind of extension to carry VPN-V4 address family routing information. This is achieved by using MBGP extension which allows BGP to carry multiprotocol address families routing information. Therefore, to deploy BGP/*MPLS* VPNs and to support the distribution of VPN-IPv4 routes, PE routers are required to support the MP-BGP extensions with VPN-IPv4 address family support.

Multiple Forwarding Tables

Each PE router needs to maintain one forwarding table per VPN that it is directly connected. These forwarding tables are called Virtual Routing and Forwarding tables (VRF). When a PE router is configured, each of it's VRF is associated with one or more *VLAN (Virtual LAN)*s, that is, there is a many to one mapping from VLAN to VPN/VRF. When receiving inbound data traffic from a directly attached CE router, PE router determines the VRF for the packet based on which VLAN the packet has been received. The ingress VLAN will uniquely map to a particular VRF in the system. Then, PE router performs a route lookup for the packet's destination IP address in the associated VRF. In the figure above, both PE1 and PE3 have two VRFs, one each for VPN-A and VPN-B, whereas, PE2 has only one VRF which is for VPN-A.

Quality of Service in BGP/MPLS VPN

The L3VPN implementation uses the existing ACL and *QoS (Quality of Service)* framework available in ExtremeXOS to achieve QoS on a PE. Please refer to the Quality of Service on page 1453 section for more information.

Virtual Routing and Forwarding Instances

The concept of Virtual Routing and Forwarding (VRF) is similar to the already present in Extreme Switches. Like a VR, each VRF also has a distinct set of VLANs and ports, as well as a unique name assigned to it across the system. VLANs are created, and ports are added to in the same manner as is done for VRs. Logically separate instances of the same routing protocol may run in the contexts of multiple VRFs of the same switch, simultaneously. In addition, a VRF may also contain a Route Distinguisher (RD), multiple import and export Router Targets (RTs), as well as a VPN ID. These three items allow a VRF to support L3VPNs.

The goal of VRF is to provide separation of overlapping address spaces belonging to separate routing domains. The objective of the VR is to split the switch into multiple routing domains so that traffic originating on one VR never enters another VR. The objective of VRFs is to control the flow of traffic across various VPN sites. As the PE-PE connection is always part of the default VR or a user VR, the forwarding table corresponding to a particular VRF has the incoming interface belonging to the VRF, and egress interfaces belonging to the default, or user VR. Similarly, the forwarding table (*MPLS* label table) corresponding to a user, or default VR, could have an interface belonging to a VRF as part of its egress interfaces. In this sense a VRF is "associated" with the user, or default VR (also refered to as parent VRs).

A VRF is not the same as a VR. A VRF always requires that you have a Parent VR, and this can either be the default VR or User VR. Protocols running in a VRF run as a separate logical instance within the context of the protocol process that is running in the Parent VR. ExtremeXOS VRFs come in two types:

- Non-VPN VRFs - Non-VPN VRFs are used for L3 Routing and Forwarding just like VRs. These provide the ability to scale protocol deployments. *BGP* and static routing are supported in the Non-VPN VRF.

- VPN VRFs - VPN VRFs enable configuration of RD and RT to realize L3VPN topologies.

  Note
  MPLS and BGP must be configured in the parent VR (PE-facing).

  BGP and static routing are supported in the VPN VRF.

Again, a VRF always requires that you have a Parent VR, and this can either be the default VR or User VR. Protocols running in a VRF run as a separate logical instance within the context of the protocol process that is running in the Parent VR. For example, the BGP process for a VR running MPLS will handle all PE to CE instances of BGP by virtualizing the data structures. This approach of allowing VRFs in a VR is more scalable than having a process for every instance of PE-to-CE routing protocol.

L3VPN BOOTP Relay

A DHCP Relay agent relays DHCP requests from the client to the DHCP server and relays the DHCP replies from the server to the client. It acts as a proxy, thereby reducing the number of DHCP servers required in the network. The DHCP relay agent inserts its own IP address in the giaddr field (gateway address) of the DHCP request. The DHCP server can look into this IP address, identify the DHCP client's subnet and assign an IP address accordingly.

You can enable BOOTP Relay over VRFs and VPN-VRFs in the same manner as with normal VRs. Normally, you must enable BOOTP Relay on both ingress and egress VLANs. However, with BOOTP Relay over L3VPN, you do not need to enable BOOTP Relay on egress VLANs, since ingress and egress VLANs are present in different VRs/VRFs.

Limitations

The following are limitations of the BOOTP Relay L3VPN feature:

- VPN-ID is not supported on DHCP-Relay Option 82, which services clients across VPNs.
- IPv6 BOOTP Relay not supported, since there is no IPv6 support for L3VPN.

L3VPN Configuration Example

Here is an example of a Layer 3 VPN. In this example CE1 and CE2 are in different AS. You can put them in the same AS by adding the `allowas-in` command.

In this example, you should be able to ping CE2 "foo1's" IP address from CE1.

Figure 116: Layer 3 Virtual Private Network

```
CE1:
configure snmp sysName "CE1"
create vlan "lo0"
enable loopback-mode vlan lo0
create vlan "vl100"
configure vlan vl100 tag 100
enable jumbo-frame ports 1
configure vlan vl100 add ports 1 tagged
configure vlan Mgmt ipaddress 192.168.56.111 255.255.255.0
configure vlan lo0 ipaddress 172.16.0.1 255.255.255.255
configure vlan vl100 ipaddress 10.1.1.1 255.255.255.252
enable ipforwarding vlan vl100

configure bgp AS-number 65100
configure bgp routerid 172.16.0.1
create bgp neighbor 10.1.1.2 remote-AS-number 65000
enable bgp neighbor 10.1.1.2
enable bgp

PE1:
configure snmp sysName "PE1"
create vr "vpn-a" type vpn-vrf vr "VR-Default"
configure vr VR-Default delete ports 1
configure vr vpn-a add ports 1
create vlan vl100 vr vpn-a tag 100
configure vl100 add ports 1 tagged
create vlan "vl101"
configure vlan vl101 tag 101
enable jumbo-frame ports 1
enable jumbo-frame ports 2
configure vlan vl100 add ports 1 tagged
configure vlan vl101 add ports 2 tagged
configure vlan Mgmt ipaddress 192.168.56.102 255.255.255.0
configure vlan lo0 ipaddress 172.16.0.2 255.255.255.255
enable ipforwarding vlan lo0
configure vlan vl101 ipaddress 10.1.1.5 255.255.255.252
enable ipforwarding vlan vl101
configure vlan vl100 ipaddress 10.1.1.2 255.255.255.252
enable ipforwarding vlan vl100

configure vr vpn-a add protocol bgp
configure vr vpn-a rd 172.16.0.2:100
configure vr vpn-a route-target both add 65000:100

enable iproute mpls-next-hop

configure bgp AS-number 65000
configure bgp routerid 172.16.0.2
create bgp neighbor 172.16.0.4 remote-AS-number 65000
```

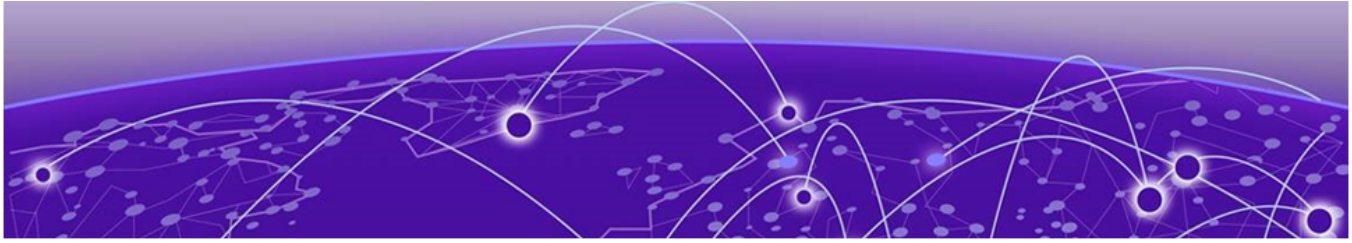
```
configure bgp neighbor 172.16.0.4 source-interface ipaddress 172.16.0.2
enable bgp neighbor 172.16.0.4
configure bgp neighbor 172.16.0.4 next-hop-self
configure bgp neighbor 172.16.0.4 address-family vpnv4 next-hop-self
enable bgp neighbor 172.16.0.4 capability vpnv4
enable bgp export vr vpn-a direct address-family vpnv4
enable bgp export vr vpn-a bgp address-family vpnv4
enable bgp

virtual-router vpn-a
configure bgp AS-number 65000
configure bgp routerid 172.16.0.2
create bgp neighbor 10.1.1.1 remote-AS-number 65100
enable bgp neighbor 10.1.1.1
disable bgp neighbor 10.1.1.1 capability ipv4-multicast
enable bgp export remote-vpn address-family ipv4-unicast
enable bgp
virtual-router VR-Default

configure mpls add vlan "lo0"
enable mpls vlan "lo0"
enable mpls ldp vlan "lo0"
configure mpls add vlan "vl101"
enable mpls vlan "vl101"
enable mpls ldp vlan "vl101"
configure mpls lsr-id 172.16.0.2
enable mpls protocol ldp
enable mpls

configure ospf routerid 172.16.0.2
enable ospf
configure ospf add vlan lo0 area 0.0.0.0 passive
configure ospf add vlan vl101 area 0.0.0.0 link-type point-to-point

P1:
configure snmp sysName "P1"

configure vlan default delete ports 1-2
create vlan "lo0"
enable loopback-mode vlan lo0
create vlan "vl101"
configure vlan vl101 tag 101
create vlan "vl102"
configure vlan vl102 tag 102
enable jumbo-frame ports 1
enable jumbo-frame ports 2
configure vlan vl101 add ports 1 tagged
configure vlan vl102 add ports 2 tagged
configure vlan Mgmt ipaddress 192.168.56.103 255.255.255.0
configure vlan lo0 ipaddress 172.16.0.3 255.255.255.255
enable ipforwarding vlan lo0
configure vlan vl101 ipaddress 10.1.1.6 255.255.255.252
enable ipforwarding vlan vl101
configure vlan vl102 ipaddress 10.1.1.9 255.255.255.252
enable ipforwarding vlan vl102

enable iproute mpls-next-hop

configure mpls add vlan "lo0"
enable mpls vlan "lo0"
enable mpls ldp vlan "lo0"
configure mpls add vlan "vl101"
enable mpls vlan "vl101"
enable mpls ldp vlan "vl101"
```

```
configure mpls add vlan "vl102"
enable mpls vlan "vl102"
enable mpls ldp vlan "vl102"
configure mpls lsr-id 172.16.0.3
enable mpls protocol ldp
enable mpls

configure ospf routerid 172.16.0.3
enable ospf
configure ospf add vlan lo0 area 0.0.0.0 passive
configure ospf add vlan vl101 area 0.0.0.0 link-type point-to-point
configure ospf add vlan vl102 area 0.0.0.0 link-type point-to-point
```

PE2:

```
configure snmp sysName "PE2"

configure vr vpn-a add ports 2
configure vr VR-Default delete ports 2
create vr "vpn-a" type vpn-vrf vr "VR-Default"
configure vlan default delete ports 2
create vlan "lo0"
enable loopback-mode vlan lo0
create vlan "vl102"
configure vlan vl102 tag 102
create vlan "vl103" vr vpn-a
configure vlan vl103 tag 103
enable jumbo-frame ports 1
enable jumbo-frame ports 2
configure vlan vl102 add ports 1 tagged
configure vlan vl103 add ports 2 tagged
configure vlan Mgmt ipaddress 192.168.56.104 255.255.255.0
configure vlan lo0 ipaddress 172.16.0.4 255.255.255.255
enable ipforwarding vlan lo0
configure vlan vl102 ipaddress 10.1.1.10 255.255.255.252
enable ipforwarding vlan vl102
configure vlan vl103 ipaddress 10.1.1.13 255.255.255.252
enable ipforwarding vlan vl103

configure vr vpn-a add protocol bgp
configure vr vpn-a rd 172.16.0.4:103
configure vr vpn-a route-target both add 65000:100

enable iproute mpls-next-hop

configure bgp AS-number 65000
configure bgp routerid 172.16.0.4
create bgp neighbor 172.16.0.2 remote-AS-number 65000
configure bgp neighbor 172.16.0.2 source-interface ipaddress 172.16.0.4
enable bgp neighbor 172.16.0.2
configure bgp neighbor 172.16.0.2 next-hop-self
configure bgp neighbor 172.16.0.2 address-family vpnv4 next-hop-self
enable bgp neighbor 172.16.0.2 capability vpnv4
enable bgp export vr vpn-a direct address-family vpnv4
enable bgp export vr vpn-a bgp address-family vpnv4
enable bgp

virtual-router vpn-a
configure bgp AS-number 65000
configure bgp routerid 172.16.0.4
create bgp neighbor 10.1.1.14 remote-AS-number 65101
enable bgp neighbor 10.1.1.14
disable bgp neighbor 10.1.1.14 capability ipv4-multicast
enable bgp export remote-vpn address-family ipv4-unicast
```

```
enable bgp
virtual-router VR-Default

configure mpls add vlan "lo0"
enable mpls vlan "lo0"
enable mpls ldp vlan "lo0"
configure mpls add vlan "vl102"
enable mpls vlan "vl102"
enable mpls ldp vlan "vl102"
configure mpls lsr-id 172.16.0.4
enable mpls protocol ldp
enable mpls

configure ospf routerid 172.16.0.4
enable ospf
configure ospf add vlan lo0 area 0.0.0.0 passive
configure ospf add vlan vl102 area 0.0.0.0 link-type point-to-point

CE2:

configure snmp sysName "CE2"

configure vr vpn-a add ports 1
configure vlan default delete ports all
configure vr VR-Default delete ports 1
configure vlan default delete ports 1
create vlan "foo1"
enable loopback-mode vlan foo1
create vlan "lo0"
enable loopback-mode vlan lo0
create vlan "vl103"
configure vlan vl103 tag 103
enable jumbo-frame ports 1
configure vlan vl103 add ports 1 tagged
configure vlan Mgmt ipaddress 192.168.56.105 255.255.255.0
configure vlan lo0 ipaddress 172.16.0.5 255.255.255.255
configure vlan vl103 ipaddress 10.1.1.14 255.255.255.252
configure vlan foo1 ipaddress 10.2.1.1 255.255.255.0
enable ipforwarding vlan foo1

configure bgp AS-number 65101
configure bgp routerid 172.16.0.5
configure bgp add network 10.2.1.0/24
create bgp neighbor 10.1.1.13 remote-AS-number 65000
enable bgp neighbor 10.1.1.13
enable bgp
```



LLDP

Supported Advertisements (TLVs) on page 1036
LLDP Redundancy Support on Stacking on page 1040
LLDP Packets on page 1041
Transmitting LLDP Messages on page 1041
Receiving LLDP Messages on page 1042
LLDP Management on page 1043
Configuring and Managing LLDP on page 1043
Displaying LLDP Information on page 1054

The Link Layer Discovery Protocol (LLDP) is defined by IEEE standard 802.1ab and provides a standard method for discovering physical network devices and their capabilities within a given network management domain.

LLDP-enabled network devices include repeaters, bridges, access points, routers, and wireless stations, and LLDP enables these devices to do the following:

- Advertise device information and capabilities to other devices in the management domain.
- Receive and store device information received from other network devices in the management domain.

LLDP-discovered information can be used to do the following:

- Discover information from all LLDP compatible devices in a multivendor environment.
- Trigger universal port profiles that can configure a switch port for a remote device (see Universal Port).
- Supply identity information that can be used for authentication and identity management.
- Provide device information to SNMP (Simple Network Management Protocol) compatible Network Management Systems such as Extreme Management Center or Ridgeline, which can present the information in inventory reports and topology maps.

The following sections provide additional information on LLDP support in the software.

Supported Advertisements (TLVs)

LLDP (Link Layer Discovery Protocol) defines a set of common advertisement messages. These are distributed in Type Length Value (TLV) format in an LLDP packet (see LLDP Packets on page 1041).

The individual advertisements within the packet are called TLVs, and each TLV advertises device information or a device capability. Certain TLVs are mandatory and are always advertised after LLDP is enabled; optional TLVs are advertised only when so enabled by default or during configuration.

The following sections provide more information on TLVs.

Mandatory TLVs

Mandatory TLVs are those TLVs that must be advertised (as defined in IEEE standard 802.1ab) when *LLDP* is enabled.

If you enable LLDP on a port, the mandatory TLVs are advertised and there is no command to disable this advertising.

Table 68: Mandatory TLVs

Name	Comments
Chassis ID	The ExtremeXOS software advertises the system's MAC address in this TLV to uniquely identify the device. **Note:** *EDP (Extreme Discovery Protocol)* also uses the MAC address to identify the device.
Port ID	The ExtremeXOS software advertises the port ID in this TLV to uniquely identify the port that sends the TLVs. This port ID is the ifName object in the MIB.
Time to live (TTL)	This TLV indicates how long the record should be maintained in the LLDP database. The default value is 120 seconds (or 2 minutes). When a port is shutting down or LLDP is disabled on a port, this TLV is set to value 0, which indicates that the record for this port should be deleted from the database on other devices. Although you cannot directly configure the TTL TLV, you can configure the transmit hold value, which is used to calculate the TTL TLV. (See Configuring LLDP Timers for more information.)
End-of-LLDP PDU	The end-of-LLDPDU (LLDP protocol data unit) TLV marks the end of the data in each LLDP packet.

Optional TLVs

IEEE standard 802.1ab defines a set of optional TLVs, which are listed in the following table.

The system description TLV is advertised by default. All other optional TLVs must be configured to enable advertising. You can use the CLI to configure the optional TLVs, or you can use an *SNMP (Simple Network Management Protocol)*-compatible network management system (NMS). For more information on the optional TLVs, see Configuring Optional TLV Advertisements.

Table 69: Optional TLVs

Name	Comments
Port description	Advertises the display-string that is configured for the port.
System name	Advertises the device's configured system name, which is configured with the configure snmp sysname command.
System description	Advertises show version command information similar to the following: `ExtremeXOS version 11.2.0.12 v1120b12 by release-manager on Fri Mar 18 16:01:08 PST 2005` The default configuration advertises this optional TLV when *LLDP* is enabled.
System capabilities	Advertises bridge capabilities. If IP forwarding is enabled on at least one *VLAN (Virtual LAN)* in the switch, the software also advertises router capabilities.
Management address	Advertises the IP address of the management VLAN in the management address TLV. If the management VLAN does not have an assigned IP address, the management address TLV advertises the system's MAC address, unless another VLAN's IP address is specified (see Management Address TLV on page 1047).
VLAN name	Advertises the name of a tagged VLAN for the port. You can configure this TLV multiple times to support multiple VLANs.
Port VLAN ID	Advertises the untagged VLAN on the port.
Port and protocol VLAN ID	Advertises a VLAN and whether the port supports protocol-based VLANs or not. You can configure this TLV multiple times to support multiple VLANs.
MAC/PHY configuration/ status	Advertises the autonegotiation and physical layer capabilities of the port.
Power via MDI	For Ethernet (*PoE (Power over Ethernet)*) or PoE+ ports, this TLV advertises the device type, power status, power class, and pin pairs used to supply power.
Link aggregation	Advertises information on the port's load-sharing (link aggregation) capabilities and status.
Maximum frame size	Advertises the maximum supported frame size for a port to its neighbors. When jumbo frames are not enabled on the specified port, the TLV advertises a value of 1518. If jumbo frames are enabled, the TLV advertises the configured value for the jumbo frames.

Avaya–Extreme Networks Optional TLVs

The software supports a set of TLVs that are proprietary to Avaya and Extreme Networks.

These TLVs advertise and receive information for Avaya voice over IP (VoIP) telephones, which include powered device (PD) information. Some TLVs are advertised by the switch, and some are advertised by the telephone. The switch starts advertising these proprietary TLVs when you enable *LLDP* and configure the specified TLVs to be advertised. The switch receives the proprietary TLVs when LLDP is enabled; you do not have to configure the switch to receive individual TLVs.

Table 70: Avaya–Extreme Networks TLVs

Name	Comments
PoE conservation level request	When the switch software needs to reduce power on a PoE-enabled port, this TLV advertises that request to the connected Avaya device.
Call server	The switch uses this TLV to advertise the IP addresses of up to eight call servers to an Avaya phone.
File server	The switch uses this TLV to advertise the IP addresses of up to four file servers to an Avaya phone.
802.1Q framing	The switch uses this TLV to advertise information about Layer 2 priority tagging to an Avaya phone.
PoE Conservation level support	An Avaya phone uses this TLV to communicate the current power consumption level and current conservation level for the phone, including typical power value, maximum power value, and available conservation power levels.
IP phone address	An Avaya phone uses this TLV to communicate the IP address and mask configured in the phone, as well as a default gateway address.
CNA server	An Avaya phone uses this TLV to communicate the IP address of a Converged Network Analyzer (CNA).
Fabric Attach (FA) Element	Used by FA elements to advertise their FA capabilities.
Fabric Attach (FA) Assignment	Used by FA clients to advertise service requests and by FA proxy and FA server elements to return the status of FA client service requests.

LLDP MED Optional TLVs

LLDP Media Endpoint Discovery (MED) is an extension to LLDP that is published as standard ANSI/TIA-1057. LLDP MED TLVs advertise and receive information for endpoint

devices, which can include powered device (PD) information. Some TLVs are advertised by the switch, and some are advertised by the endpoint device.

Table 71: LLDP MED TLVs

Name	Comments
LLDP MED capabilities	The switch uses this TLV to advertise the switch LLDP MED capabilities to endpoint devices. This TLV must be configured for advertisement before any other LLDP MED TLVs can be configured for advertisement, and advertisement for all other MED TLVs, must be disabled before advertisement for this TLV can be disabled.
Network policy	The switch uses this TLV to advertise network policy information for specific applications to endpoint devices. **Note:** Network policies cannot be configured by *SNMP*-based management programs.
Location ID	The switch uses this TLV to advertise a location ID to an endpoint device.
Extended power via MDI	The switch uses this TLV to advertise a power information and settings to an endpoint device on a *PoE*-capable port.
Hardware revision	An endpoint device uses this TLV to advertise the configured hardware revision to the switch.
Firmware revision	An endpoint device uses this TLV to advertise the configured firmware revision to the switch.
Software revision	An endpoint device uses this TLV to advertise the configured software revision to the switch.
Serial number	An endpoint device uses this TLV to advertise the configured serial number to the switch.
Manufacturer name	An endpoint device uses this TLV to advertise the configured manufacturer name to the switch.
Model name	An endpoint device uses this TLV to advertise the configured model name to the switch.
Asset ID	An endpoint device uses this TLV to advertise the configured asset ID to the switch.

You must enable the LLDP-MED capabilities TLV for advertising before you configure any other LLDP MED TLVs for advertising. Likewise, when disabling LLDP MED TLV advertisement, you can disable the LLDP-MED capabilities TLV only after you have disabled advertisement for all other LLDP MED TLVs.

After the LLDP-MED capabilities TLV is configured for advertising, the switch can receive LLDP MED TLVs from endpoints; you do not have to configure the switch to receive individual TLVs.

The switch advertises LLDP MED TLVs only after the switch receives an LLDP MED TLV from an endpoint, and the switch only advertises on ports from which an LLDP MED TLV has been received. This approach prevents the switch from advertising LLDP

MED TLVs to another switch, and it prevents the wasted bandwidth and processing resources that would otherwise occur.

The LLDP MED protocol extension introduces a new feature called MED fast start, which is automatically enabled when the LLDP MED capabilities TLV is configured for advertisement.

When a new LLDP MED-capable endpoint is detected, the switch advertises the configured LLDP MED TLVs every one second for the configured number of times (called the repeat count). This speeds up the initial exchange of LLDP MED capabilities. After the repeat count is reached, the LLDP MED TLVs are advertised at the configured interval.

Note
The fast-start feature is automatically enabled, at the default level of 3, when you enable the LLDP MED capabilities TLV on the port.

BGP Config Discovery Optional TLV

LLDP BGP (Border Gateway Protocol) Peer Discovery is an extension to LLDP from the IETF BGP LLDP Peer Discovery draft. The TLVs advertise BGP information that is used to discover BGP peers. The TLV is advertised when Auto-peering is created. LLDP BGP Config consists of a single TLV with multiple sub-TLVs within the TLV.

Table 72: BGP Config Discovery Optional TLV

Name	Comments
Peering address	Advertises the local IP addresses used for BGP sessions. When Auto-peering is created, this advertises the IPv6 link-local IP address of the system.
BGP local AS	Advertises the BGP local Autonomous System (AS) numbers. When Auto-peering is created, this advertises Auto-peering's AS-number.
BGP identifier	Advertises the local BGP identifier. When Auto-peering is created, this advertises Auto-peering's router ID.
Session group-ID	Advertises a value to identify a BGP session class or category.
BGP session capabilities	Advertises the capabilities of the BGP session.
Key chain	Advertises the key chain name used for BGP session authentication.

LLDP Redundancy Support on Stacking

ExtremeXOS 31.6 introduces active redundancy support for LLDP on stacking.

LLDP information (for example, LLDP neighbors, Fabric Attach Elements and Assignments) is continuously synchronized from the Primary to the Backup nodes. In case of Failover, the Backup becomes the Primary node, with all existing LLDP information retained. Existing LLDP and Fabric Attach neighbors are not re-learned.

Because Fabric Attach Assignments are retained, there is no impact on the traffic during Failover.

Note
Statistics information is not synchronized to the Backup. LLDP and Fabric Attach statistics start from empty on Failover.

LLDP Packets

LLDP packets transport TLVs to other network devices (see Figure 117).

The LLDP packet contains the destination multicast address, the source MAC address, the LLDP EtherType, the LLDPDU data (which contains the TLVs), and a frame check sequence (FCS). The LLDP multicast address is defined as 01:80:C2:00:00:0E, and the EtherType is defined as 0x88CC.

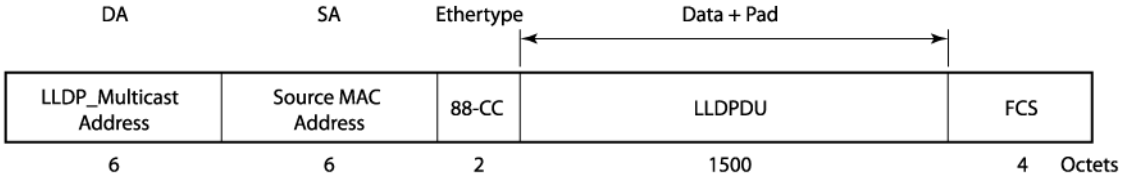


Figure 117: LLDP Packet Format

The following characteristics apply to LLDP packets:

- They are IEEE 802.3 Ethernet frames.
- The frames are sent as untagged frames.
- The frames are sent with a link-local-assigned multicast address as the destination address.
- The *STP (Spanning Tree Protocol)* state of the port does not affect the transmission of LLDP frames.

The length of the packet cannot exceed 1500 bytes. As you add TLVs, you increase the length of the LLDP frame. When you reach 1500 bytes, the remaining TLVs are dropped. We recommend that you advertise information regarding only one or two VLANs on the LLDP port, to avoid dropped TLVs.

If the system drops TLVs because of exceeded length, the system logs a message to the EMS and the `show lldp statistics` command shows this information under the Tx Length Exceeded field.

Note
The LLDPDU maximum size is 1500 bytes, even with jumbo frames enabled. TLVs that exceed this limit are dropped.

Transmitting LLDP Messages

You can configure each port to transmit *LLDP* messages, receive LLDP messages, or both.

When configured to transmit LLDP messages, the LLDP agent running on the switch passes serially through the list of ports that are enabled for LLDP. For each LLDP-enabled port, the switch periodically sends out an untagged LLDP packet that contains the mandatory LLDP TLVs as well as the optional TLVs that are configured for advertisement. These TLVs are advertised to all neighbors attached to the same network. LLDP agents cannot solicit information from other agents by way of this protocol.

Note

When both IEEE 802.1X and LLDP are enabled on the same port, LLDP packets are not sent until one or more clients authenticate a port.

Also, LLDP MED TLVs are advertised only after an LLDP MED TLV is received on a port that is configured for LLDP MED. (See LLDP MED Optional TLVs on page 1038.)

The source information for TLVs is obtained from memory objects such as standard MIBs or from system management information. If the TLV source information changes at any time, the LLDP agent is notified. The agent then sends an update with the new values, which is referred to as a triggered update. If the information for multiple elements changes in a short period, the changes are bundled together and sent as a single update to reduce network load.

Receiving LLDP Messages

You can configure each port to transmit *LLDP* messages, receive LLDP messages, or both.

After you configure a port to receive TLVs, all LLDP TLVs are received (even if the LLDP MED capabilities TLV is not enabled). Each port can store LLDP information for a maximum of four neighbors.

Note

When both IEEE 802.1X and LLDP are enabled on the same port, incoming LLDP packets are accepted only when one or more clients are authenticated.

When configured to receive LLDP messages, the TLVs received at a port are stored in a standard Management Information Base (MIB), which makes it possible for the information to be accessed by an *SNMP*-compatible NMS. Unrecognized TLVs are also stored on the switch, in order of TLV type. TLV information is purged after the configured timeout interval, unless it is refreshed by the remote LLDP agent. You can also manually clear the LLDP information for one or all ports with the `clear lldp neighbors` command.

- To display TLV information received from LLDP neighbors, use the following command:

  ```
  show lldp neighbors detailed
  ```

  You must use the detailed variable to display all TLV information.

LLDP Management

You can manage *LLDP* using the CLI and/or an *SNMP*-compatible NMS. LLDP works concurrently with the *EDP*. It also works independently; you do not have to run EDP to use LLDP.

You can use the `save configuration` command to save LLDP configurations across reboots.

The switch logs EMS messages regarding LLDP, including when optional TLVs exceed the 1500-byte limit and when more than four neighbors are detected on a port.

After you enable LLDP, you can enable LLDP-specific SNMP traps; the traps are disabled by default. After you enable LLDP-specific traps, the switch sends all LLDP traps to the configured trap receivers. You can configure the period between SNMP notifications; the default interval is five seconds.

You can configure an optional TLV to advertise or not advertise the switch management address information to the neighbors on a port. When enabled, this TLV sends out the IPv4 address configured on the management *VLAN*. If you have not configured an IPv4 address on the management VLAN, the software advertises the system's MAC address. LLDP does not send IPv6 addresses in this field.

Configuring and Managing LLDP

The following sections describe how to configure *LLDP* on the switch.

Configuration Overview

You configure LLDP per port, and each port can store received information for a maximum of four neighbors.

Note
LLDP runs with link aggregation

You can configure LLDP per port.

1. Enable LLDP on the desired port(s) as described in Enabling and Disabling LLDP.
2. If you want to change any default timer values, see Configuring LLDP Timers.
3. Enable the SNMP traps and configure the notification interval as described in Configuring SNMP for LLDP.
4. Configure any optional TLV advertisements as described in Configuring Optional TLV Advertisements.

Note
By default, an LLDP-enabled port advertises the optional system description TLV. By default, all other optional TLVs are not advertised.

Enabling and Disabling LLDP

LLDP is enabled on all ports by default. When you enable LLDP on one or more ports, you select whether the ports will only transmit LLDP messages, only receive the messages, or both transmit and receive LLDP messages.

After you enable LLDP, the following TLVs are automatically added to the LLDPDU:

- Chassis ID
- Port ID
- TTL
- System description
- End of LLDPDU

All of these, except the system description, are mandated by the 802.1ab standard and cannot be configured. For information on changing the system description TLV advertisement, see System Description TLV on page 1046.

- Enable LLDP.

  ```
  enable lldp ports [all | port_list] {receive-only | transmit-only}
  ```

- Disable LLDP.

  ```
  disable lldp ports [all | port_list] {receive-only | transmit-only}
  ```

Configuring LLDP Timers

The *LLDP* timers apply to the entire switch and are not configurable by port. After you enable LLDP, LLDP timers control the time periods for the transmission and storage of LLDP TLVs as follows:

- Reinitialization period (default is two seconds).
- Delay between LLDP transmissions (default is two seconds)—applies to triggered updates, or updates that are initiated by a change in the topology.
- Transmit interval (default is 30 seconds)—applies to messages sent periodically as part of protocol.
- Time-to-live (TTL) value (default is two minutes)—time that the information remains in the recipient's LLDP database.

Note

Once the LLDP MED TLVs begin transmitting (after detecting LLDP MED TLVs from a connected endpoint), those TLVs are also controlled by these timers.

When LLDP is disabled or if the link goes down, LLDP is reinitialized. The reinitialize delay is the number of seconds the port waits to restart the LLDP state machine; the default is two seconds.

The time between triggered update LLDP messages is referred to as the transmit delay, and the default value is two seconds. You can change the default transmit delay value to a specified number of seconds or to be automatically calculated by multiplying the transmit interval by 0.25.

Each LLDP message contains a TTL value. The receiving LLDP agent discards all LLDP messages that surpass the TTL value; the default value is 120 seconds. The TTL is calculated by multiplying the transmit interval value and the transmit hold value; the default transmit hold value is four.

- To change the default reinitialize delay period, use the following command:

  configure lldp reinitialize-delay seconds

  LLDP messages are transmitted at a set interval; this interval has a default value of every 30 seconds.
- To change this default value, use the following command:

  configure lldp transmit-interval seconds
- To change the value for the transmit delay, use the following command:

  `configure lldp transmit-delay [ auto | seconds]`
- To change the default transmit hold value, use the following command:

  `configure lldp transmit-hold hold`

Configuring SNMP for LLDP

By default, *SNMP LLDP* traps are disabled on all ports. The default value for the interval between SNMP LLDP trap notifications is 5 seconds.

- To enable LLDP SNMP traps on one or more ports, use the following command:

  `enable snmp traps lldp {ports [all | port_list]}`

  The traps are only sent for those ports that are both enabled for LLDP and have LLDP traps enabled.
- To disable the LLDP SNMP traps on one or more ports, use the following command:

  `disable snmp traps lldp {ports [all | port_list]}`
- To change the default value for the interval for the entire switch, use the following command:

  `configure lldp snmp-notification-interval seconds`

  Note

  If you want to send traps for LLDP MED, you must configure it separately. Use the `enable snmp traps lldp-med {ports [all | port_list]}` command to enable these traps.

Configuring Optional TLV Advertisements

By default, the ExtremeXOS software advertises the mandatory *LLDP* TLVs (which are not configurable) and the optional system description TLV. For details, see Supported Advertisements (TLVs) on page 1036). All other optional TLVs are not advertised.

You can choose to advertise or not advertise any optional TLV, but be aware that the total LLDPDU length, which includes the mandatory TLVs, cannot exceed 1500 bytes. Optional TLVs that cause the LLDPDU length to exceed the 1500-byte limit are

dropped. You can see if the switch has dropped TLVs by referring to the EMS log or by issuing the `show lldp statistics` command.

The following sections describe configuration for the following types of optional TLVs.

Configuring Standards-Based TLVs

This section describes the following optional standards-based TLVs.

Port description TLV

The port description TLV advertises the ifDescr MIB object, which is the ASCII string you configure.

The string can be configured using the `configure ports display-string` command.

If you have not configured this parameter, the TLV carries an empty string.

To control advertisement of the port description TLV, use the following command:

`configure lldp ports [all | port_list] [advertise | no-advertise] port-description`

System name TLV

The system name TLV advertises the configured system name for the switch. This is the sysName as defined in RFC 3418, which you can define using the `configure snmp sysname` command.

To control advertisement of the system name TLV, use the following command:

`configure lldp ports [all | port_list] [advertise | no-advertise] system-name`

System Description TLV

By default, the ExtremeXOS software advertises this TLV whenever you enable *LLDP* on a port, but you can disable advertisement. This TLV advertises show version command information similar to the following in the system description TLV:

```
ExtremeXOS version 11.2.0.12 v1120b12 by release-manager
on Fri Mar 18 16:01:08 PST 2005
```

To control advertisement of the system description TLV, use the following command:

`configure lldp ports [all | port_list] [advertise | no-advertise] system-description`

System Capabilities TLV

The system capabilities TLV advertises the capabilities of the switch and which of these capabilities are enabled. When so configured, the ExtremeXOS software advertises bridge capabilities. If IP forwarding is enabled on at least one *VLAN* in the switch, the software also advertises router capabilities.

To control advertisement of the system capabilities TLV, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
system-capabilities
```

Management Address TLV

The management address TLV advertises the IP address of the management *VLAN*. If the management VLAN does not have an assigned IP address, the management address TLV advertises the system's MAC address, unless another VLAN's IP address is specified. *LLDP* does not recognize IPv6 addresses in this field.

To specify a VLAN IP address, other than the Management VLAN, for advertisement as the management address, use the following command:

```
configure lldp management-address [[[{vlan} vlan_name | vlan vlan_id]
{primary-ip | secondary-ip secondary_ip_address}] | mac-address]
```

To turn on or off advertisement of the management address TLV, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
management-address
```

Note

The ExtremeXOS software sends only one management address.

VLAN Name TLV

The *VLAN* name TLV advertises a VLAN name for one or more VLANs on the port. You can advertise this TLV for tagged and untagged VLANs. When you enable this TLV for tagged VLANs, the TLV advertises the IEEE 802.1Q tag for that VLAN. For untagged VLANs, the internal tag is advertised.

If you do not specify a VLAN when you configure this TLV, the switch advertises all VLAN names on the specified ports. You can choose to advertise one or more VLANs for the specified ports by specifying the name of a VLAN in the configuration command. You can repeat the command to specify multiple VLANs.

To control advertisement of the port VLAN Name TLV, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific dot1 vlan-name {vlan [all | vlan_name]}
```

Note

Because each VLAN name requires 32 bits and the LLDPDU cannot exceed 1500 bytes, we recommend that you configure each port to advertise no more than one or two specific VLANs. Optional TLVs that cause the LLDPDU length to exceed the 1500-byte limit are dropped.

Port VLAN ID TLV

The port *VLAN* ID advertises the untagged VLAN on that port. Thus, only one port VLAN ID TLV can exist in the LLDPDU. If you configure this TLV and there is no untagged VLAN on the specified port, this TLV is not included in the LLDPDU.

- To control advertisement of the port VLAN ID TLV, use the following command:
  `configure lldp ports [all | port_list] [advertise | no-advertise] vendor-specific dot1 port-vlan-ID`

Port and Protocol VLAN ID TLV

When configured for advertisement, this TLV advertises whether the specified *VLAN*s on the specified ports support protocol-based VLANs or not.

If you do not specify a VLAN when you configure this TLV, the switch advertises protocol-based VLAN support for all VLAN names on the specified ports. You can choose to advertise support for one or more VLANs for the specified ports by specifying the name of a VLAN in the configuration command. You can repeat the configuration command to specify multiple VLANs.

Because all VLANs on Extreme Networks switches support protocol-based VLANs, the switch always advertises support for protocol-based VLANs for all VLANs for which this TLV is advertised. If no protocol-based VLANs are configured on the port, the switch sets the VLAN ID value to 0.

- To control advertisement of the port and protocol VLAN ID TLV, use the following command:
  `configure lldp ports [all | port_list] [advertise | no-advertise] vendor-specific dot1 port-protocol-vlan-ID {vlan [all | vlan_name]}`

Note

Because a TLV is advertised for every VLAN that is advertised, and because the LLDPDU cannot exceed 1500 bytes, we recommend that you advertise this VLAN capability only for those VLANs that require it. Optional TLVs that cause the LLDPDU length to exceed the 1500-byte limit are dropped.

MAC/PHY Configuration/Status TLV

When configured for advertisement, this TLV advertises the autonegotiation and physical layer capabilities of the port. The switch adds information about the port speed, duplex setting, bit rate, physical interface, and autonegotiation support and status.

- To control advertisement of the port and protocol MAC/PHY configuration/status TLV, use the following command:
  `configure lldp ports [all | port_list] [advertise | no-advertise] vendor-specific dot3 mac-phy`

Power Via MDI TLV

When configured for advertisement on Ethernet (*PoE*) or PoE+ ports, this TLV advertises the device type, power status, power class, and pin pairs used to supply power.

The device type field contains a binary value that represents whether the *LLDP*-compatible device transmitting the LLDPDU is a power sourcing entity (PSE) or powered device (PD), as listed in Table 73.

Table 73: Power Management TLV Device Information

Value	Power source
0	PSE device
1	PD device
2-3	Reserved

Control advertisement of the power via MIDI TLV.

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific dot3 power-via-mdi {with-classification}
```

Refer to for Configuring Avaya-Extreme TLVs on page 1049 and Configuring LLDP MED TLVs on page 1051 more information on power-related TLVs.

Link Aggregation TLV

When configured for advertisement, this TLV advertises information on the port's load-sharing (link aggregation) capabilities and status.

To control advertisement of the link aggregation TLV, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific dot3 link-aggregation
```

Maximum frame size TLV

When configured for advertisement, this TLV advertises the maximum supported frame size for a port to its neighbors. When jumbo frames are not enabled on the specified port, the TLV advertises a value of 1518. If jumbo frames are enabled, the TLV advertises the configured value for the jumbo frames.

To control advertisement of the maximum frame size TLV, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific dot3 max-frame-size
```

Configuring Avaya-Extreme TLVs

This section describes the following optional proprietary Avaya-Extreme Networks TLVs that you can configure the switch to transmit.

Note

You can display the values for these TLVs using the `show lldp neighbors detailed` command.

PoE Conservation Level Request TLV

When configured for advertisement, this TLV advertises a request to the connected PD to go into a certain power conservation level or go to the maximum conservation level. This *LLDP* TLV is sent out only on *PoE*-capable Ethernet ports.

By default, the requested conservation value on this proprietary LLDP TLV is 0, which is no power conservation. You can change this level temporarily using a network station or *SNMP* with the MIB; this change is not saved across a reboot.

- To control advertisement of the PoE conservation level request TLV, use the following command:

  ```
  configure lldp ports [all | port_list] [advertise | no-advertise]
  vendor-specific avaya-extreme poe-conservation-request
  ```

Call Server TLV

When configured for advertisement, this TLV advertises the IP addresses of up to eight call servers. Avaya phones use this addressing information to access call servers.

- To control advertisement of the call server TLV and define call server addresses, use the following command:

  ```
  configure lldp ports [all | port_list] [advertise | no-advertise]
  vendor-specific avaya-extreme call-server ip_address_1 {ip_address_2
  {ip_address_3 {ip_address_4 {ip_address_5 {ip_address_6 {ip_address_7
  {ip_address_8}}}}}}
  ```

File Server TLV

When configured for advertisement, this TLV advertises the IP addresses of up to 4 file servers. Avaya phones use this address information to access file servers.

- To control advertisement of the file server TLV and define file server addresses, use the following command:

  ```
  configure lldp ports [all | port_list] [advertise | no-advertise]
  vendor-specific avaya-extreme file-server ip_address_1 {ip_address_2
  {ip_address_3 {ip_address_4}}}
  ```

802.1Q Framing TLV

When configured for advertisement, this TLV advertises information about Layer 2 priority tagging for Avaya phones.

- To control advertisement of the 802.1Q framing TLV, use the following command:

  ```
  configure lldp ports [all | port_list] [advertise | no-advertise]
  vendor-specific avaya-extreme dot1q-framing [tagged | untagged | auto]
  ```

  Note

  For this command to work, you must have previously enabled both the `configure lldp ports vendor-specific med capabilities` and the `configure lldp ports vendor-specific med policy application` commands. (See Configuring LLDP MED TLVs on page 1051 for complete information.)

Configuring LLDP MED TLVs

Note

After you enable an *LLDP* MED TLV, the switch waits until it detects a MED-capable device before it begins transmitting the configured LLDP MED TLVs.

You must configure the LLDP MED capabilities TLV to advertise before any of the other LLDP MED TLVs can be configured. Also, this TLV must be set to no-advertise after all other MED TLVs are set to no-advertise.

This approach assures that network connectivity devices advertise LLDP MED TLVs only to end-devices and not to other network connectivity devices.

The following sections describe LLDP MED TLVs and features.

Note

You can display the values for these TLVs using the `show lldp neighbors detailed` command.

LLDP MED capabilities TLV

This TLV advertises the *LLDP* MED capabilities of the switch and must be enabled before any of the other LLDP MED TLVs can be enabled.

To enable configuration and transmission of any other LLDP MED TLV and to determine the LLDP MED capabilities of endpoint devices, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific med capabilities
```

LLDP MED Fast-Start Feature

The *LLDP* MED fast-start feature allows you to increase the learning speed of the switch for LLDP MED TLVs. The fast-start feature is automatically enabled once you enable the LLDP MED capabilities TLV.

By default, the switch sends out the LLDPDU every second for up to the default repeat count, which is 3. Once the repeat count is reached, the configured transmit interval value is used between LLDPDUs. You can configure the repeat count to any number in the range of 1 to 10.

To configure the LLDP fast-start feature, use the following command:

```
configure lldp med fast-start repeat-count count
```

Network policy TLV

This TLV advertises which *VLAN* an endpoint should use for the specified application. You can configure only one instance of an application on each port, and you can configure a maximum of eight applications, each with its own DSCP value and/or priority tag.

To control advertisement and configure one or more network policy TLVs for a port, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific med policy application [voice | voice-signaling |guest-
voice | guest-voice-signaling | softphone-voice | video-conferencing
| streaming-video | video-signaling] vlan vlan_name dscp dscp_value
{priority-tagged}
```

Location identification TLV

This TLV advertises one of three different location identifiers for a port, each with a different format.

- Coordinate-based, using a 16-byte hexadecimal string.
- Civic-based, using a hexadecimal string with a minimum of six bytes.
- ECS ELIN, using a numerical string with a range of 10-25 characters.

To control advertisement and configure location information, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific med location-identification [coordinate-based hex_value
| civic-based hex_value | ecs-elin elin]
```

Extended power-via-MDI TLV

This TLV advertises fine-grained power requirement details, including the *PoE* settings and support. You can enable this TLV only on PoE-capable ports; the switch returns an error message if you attempt to transmit this *LLDP* TLV over a non-PoE-capable port.

To control advertisement for this TLV, use the following command:

```
configure lldp ports [all | port_list] [advertise | no-advertise]
vendor-specific med power-via-mdi
```

SNMP Traps for LLDP MED

To receive *SNMP* traps on the *LLDP* MED, you must enable these separately from the other LLDP traps. (See Configuring SNMP for LLDP.)

- Enable the LLDP MED SNMP traps.

  ```
  enable snmp traps lldp-med {ports [all | port_list]}
  ```

- Disable the LLDP MED SNMP traps.

  ```
  disable snmp traps lldp-med {ports [all | port_list]}
  ```

Clearing LLDP Neighbor Entries

- To remove the *LLDP* entries received on one or more ports from the switch database, use the following command:

  ```
  clear lldp neighbors [all | port port_list]
  ```

Unconfiguring LLDP

- To unconfigure the *LLDP* timers, use the following command:

  ```
  unconfigure lldp
  ```

  This command returns the LLDP timers to default values; LLDP remains enabled, and all the configured TLVs are still advertised.
- To leave LLDP enabled, but reset the advertised TLVs to the five default TLVs, use the following command, and specify the affected ports:

  ```
  unconfigure lldp port [all | port_list]
  ```

Instant Port Enhancements

Version 33.2.1 adds the Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) Voice VLAN options to the VLAN part of Instant Port Profile. This adds the following feature functionality:

- LLDP Voice VLAN options:
  - Enable or disable LLDP advertisement of 802.1 VLAN ID and port protocol of VLAN.
  - Enable or disable med Voice VLAN DSCP:
    - Med VLAN DSCP Value
  - Enable or disable LLDP advertisement of med Voice Signaling VLAN DSCP:
    - VLAN DSCP Value
- CDP Voice VLAN Options:
  - Enable or disable CDP advertisement of VLAN.
  - Enable or disable CDP advertisement of power available.

Version 33.5.1 enhances Instant Port functionality to support multi-link detection between switches without requiring LAG configuration. This update enables Instant Port to correctly identify multiple physical links between switches—even when one or more links are in STP-blocked state—by matching the same MAC address across ports.

Key enhancements include:

- **Support for non-LAG multi-link detection**: Instant Port can now detect multiple links between switches using MAC address correlation, even when STP blocks one of the paths.
- **Port-based VLAN assignment**: A new global CLI attribute `untagged-vlan-action-mode` allows Instant Port to apply VLANs based on port identity rather than MAC address. This is critical for scenarios where MAC-based VLAN assignment is insufficient due to identical MACs across multiple links.
- **New Device Type – `Extreme Networks Switch`**: Added to Instant Port profiles to facilitate interswitch detection using LLDP system description matching. This enables automatic uplink configuration using the `Interlink` action when a neighboring switch is detected.

The following new command supports this feature enhancement:

`configure instant-port` **`untagged-vlan-action-mode`** [**`port-based`** | **`mac-based`**]

Version 33.7.1 adds an on-demand rematch operation for Instant Port Profile (IPP), allowing matched devices to be re-evaluated against the current profile configuration.

Enhancement: Instant Port Profile matches connected devices to a profile device-type based on the best available information at the time of matching. Because matched devices remain associated with their original device-type even after switch state changes — such as a profile update adding new device-types, or a delayed LLDP neighborship that would have produced a better match than an initial MAC-learning match — the matched state can become inconsistent over time. The new **`run instant-port profile rematch`** command re-evaluates all currently matched devices and associates each with the most appropriate device-type based on current switch state.

You can use the following command to trigger an on-demand rematch of all matched IPP devices:

```
run instant-port profile rematch
```

Supported Platforms

All platforms.

Displaying LLDP Information

The following sections describe how to display *LLDP* information for the switch. The system displays information on the LLDP status and statistical counters of the ports, as well as about the LLDP advertisements received and stored by the system.

You can display information on the LLDP port configuration and on the LLDP neighbors detected on the port.

Displaying LLDP Port Configuration Information and Statistics

- To display *LLDP* port configuration information, use the following command:

  `show lldp {port [all | port_list]} {detailed}`
- To display the statistical counters related to the LLDP port, use the following command:

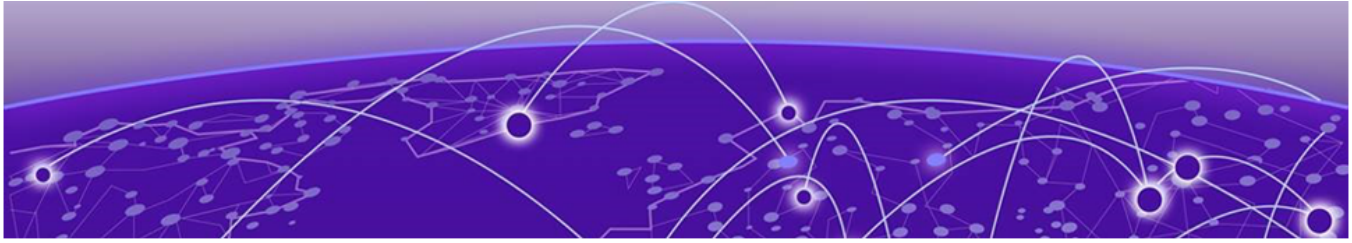
  `show lldp {port [all | port_list]} statistics`

Display LLDP Information Collected from Neighbors

- Display information collected from *LLDP* neighbors.

  ```
  show lldp neighbors
  ```

  Note
  You must use the detailed option to display information on the proprietary Avaya-Extreme Networks TLVs and the LLDP MED TLVs.



MPLS

MPLS Overview on page 1056
Configuring MPLS on page 1106
Displaying MPLS Configuration Information on page 1117
MPLS Configuration Example on page 1124
Configuring MPLS Layer-2 VPNs (VPLS and VPWS) on page 1126
VPLS VPN Configuration Examples on page 1131
Configuring H-VPLS on page 1134
Configuring Protected VPLS on page 1135
Configuring RSVP-TE on page 1135
RSVP-TE Configuration Example on page 1142
Troubleshooting MPLS on page 1145

Multiprotocol Label Switching (MPLS) is a connection-oriented technology that allows routers to make protocol-independent forwarding decisions based on fixed-length labels. This chapter provides an overview and discusses how to configure, monitor and troubleshoot the MPLS feature. The chapter provides specific configuration examples.

MPLS Overview

MPLS (Multiprotocol Label Switching) provides advanced IP services for switches that contain advanced ASICs that support MPLS.

To configure MPLS, L2 VPNs, and/or L3 VPNs on your switch, you need the MPLS enabled feature pack license .

Note
MPLS and MPLS subfeatures are supported on the platforms listed for specific features in the Switch Engine v33.6.1 Licensing Guide document.

Note
ExtremeXOS MPLS does not support Graceful Restart. Any restart of the MPLS process or failover to a backup node requires re-signaling of all MPLS-related connections and entities.

How MPLS Works

MPLS is a connection-oriented technology that allows routers to make protocol-independent forwarding decisions based on fixed-length labels.

The use of MPLS labels enables routers to avoid the processing overhead of delving deeply into each packet and performing complex route lookup operations based upon destination IP addresses.

MPLS protocols are designed primarily for routed IP networks and work together to establish multiple, unidirectional Label Switched Path (LSP) connections through an MPLS network. Once established, an LSP can be used to carry IP traffic or to tunnel other types of traffic, such as bridged MAC frames. The tunnel aspects of LSPs, which are important in supporting virtual private networks (VPNs), result from the fact that forwarding is based solely on labels and not on any other information carried within the packet.

The MPLS protocols operate on Label Switch Routers (LSRs). The router where an LSP originates is called the ingress LSR, while the router where an LSP terminates is called the egress LSR. Ingress and egress LSRs are also referred to as Label Edge Routers (LERs). For any particular LSP, a router is either an ingress LER, an intermediate LSR, or an egress LER. However, a router may function as an LER for one LSP, while simultaneously functioning as an intermediate LSR for another LSP.

Figure 118 illustrates an MPLS network.

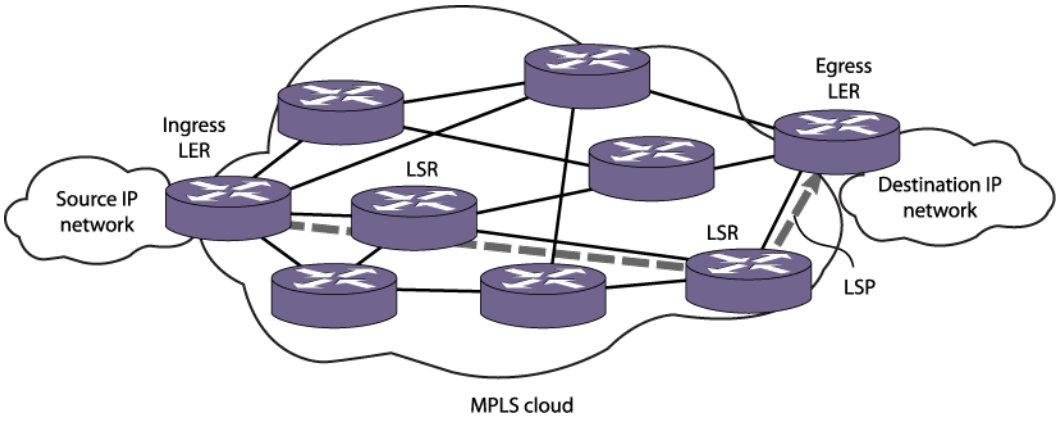


Figure 118: MPLS Network

In an MPLS environment, incoming packets are initially assigned labels by the ingress LER. The labels allow more efficient packet handling by MPLS-capable routers at each point along the forwarding path.

An MPLS label essentially consists of a short fixed-length value carried within each packet header that identifies a Forwarding Equivalence Class (FEC). The FEC tells the router how to handle the packet. An FEC is defined to be a group of packets that are forwarded in the same manner. Examples of FECs include an IP prefix, a host address, or a *VLAN (Virtual LAN)* ID.

Note

The label concept in MPLS is analogous to other connection identifiers, such as an ATM VPI/VCI or a Frame Relay DLCI.

By mapping to a specific FEC, the MPLS label efficiently provides the router with all of the local link information needed for immediate forwarding to the next hop. MPLS creates an LSP along which each LSR can make forwarding decisions based solely upon

the content of the labels. At each hop, the LSR simply strips off the Incoming label and applies a new Outgoing label that tells the next LSR how to forward the packet. This allows packets to be tunneled through an IP network.

MPLS Protocol Preference

When LSPs from different protocols exist to the same destination/FEC, LSPs from only one protocol are used at a time for that destination/FEC. The preference, or precedence, for each LSP user is defined as follows.

For both IP/L3VPN and L2VPN the order of preference is:

- RSVP-TE
- LDP
- static

MPLS Terms and Acronyms

Table 74 defines common *MPLS* terms and acronyms.

Table 74: MPLS Terms and Acronyms

Term or Acronym	Description
CSPF	Constrained Shortest Path First. Route selection determined by an algorithm based on available link bandwidth and path cost.
DoD	Downstream-on-Demand. Distribution of labels as a result of explicit upstream label requests.
DU	Downstream Unsolicited. Distribution of labels downstream without an explicit label request.
EXP bits	A three-bit experimental field in an MPLS shim header.
FEC	Forward Equivalence Class. A group of packets that are forwarded in the same manner (for example, over the same Label Switched Path).
Label	A short, fixed-length identifier used to forward packets from a given link.
Label stack	A set of one or more MPLS labels used by MPLS to forward packets to the appropriate destination.
Label swapping	Lookup and replacement of an incoming label with the appropriate outgoing label.
LDP	Label Distribution Protocol. A protocol defined by the IETF used to establish an MPLS Label Switched Path (LSP).
LER	Label Edge Router. A Label Switch Router that is at the beginning (ingress) or end (egress) of an LSP.
LSP	Label Switched Path. The unidirectional MPLS connection between two routers over which packets are sent. LSPs are established using LDP or RSVP-TE.

Table 74: MPLS Terms and Acronyms (continued)

Term or Acronym	Description
LSR	Label Switch Router. A router that receives and transmits packets on an MPLS network.
MPLS	MultiProtocol Label Switching. A set of protocols defined by the IETF used to transmit information based on a label-switching forwarding algorithm.
NHLFE	Next Hop Label Forwarding Entry. The NHLFE represents the MPLS router next hop along the LSP.
PHP	Penultimate Hop Popping. A label stack optimization used for conserving the number of allocated labels.
PW	Pseudowire. A logical point-to-point connection.
RSVP	Resource ReSerVation Protocol (RSVP). A resource setup protocol designed for an integrated services network.
RSVP-TE	Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE). The combination of RSVP and MPLS label signaling to provide traffic engineered LSPs as specified in RFC 3209RFC 3209, RSVP-TE: Extensions to RSVP for LSP Tunnels.
Shim header	MPLS-specific header information that is inserted between layer-2 and layer-3 information in the data packet.
SP	Service Provider. An entity that provides network services for individuals or organizations.
TE	Traffic Engineering. The provisioning of an autonomous flow along a specified network path.
Transport LSP	Any active LSP used to forward traffic through an MPLS network.
VPLS	Virtual Private LAN Service (VPLS). A multipoint Layer 2 VPN service that has the property that all PW tunnels within a VPN are signaled with the same vcid, where the vcid represents the VPN identifier.
VPN	Virtual Private Network (VPN). A logical private network domain that spans a public or service provider network infrastructure.
VPWS	Virtual Private Wire Service (VPWS). A point-to-point Layer 2 VPN service that operates over MPLS.

LDP Support

The Label Distribution Protocol (LDP) is a protocol defined by the IETF for the purpose of establishing *MPLS* LSPs. Using LDP, peer LSRs exchange label binding information to create LSPs. The LDP features supported in this release include:

- Downstream unsolicited label advertisement.
- Liberal label retention.
- Ordered control mode.
- Advertisement of labels for direct interfaces, *RIP (Routing Information Protocol)* routes, and static routes.

- Ability to use multiple IGP routing protocols (for example IS-IS, *OSPF (Open Shortest Path First)*, and *BGP (Border Gateway Protocol)*).
- LDP loop detection.
- Configurable LDP timers.

Note
To use BGP as an IGP routing protocol, issue the `enable mpls ldp bgp-routes` command.

LDP Neighbor Discovery

LDP includes a neighbor discovery protocol that runs over UDP.

Using the basic discovery mechanism, each LSR periodically multicasts a hello message to a well-known UDP port to which all LSRs listen. These hello messages are transmitted to the all routers on this subnet multicast group. When a neighbor is discovered, a hello-adjacency is formed and the LSR with the numerically greater IP address is denoted as the active LSR.

Hello messages must continue to be received periodically for the hello-adjacency to be maintained. The hold time that specifies the duration for which a hello message remains valid can be negotiated by the peer LSRs as part of the HELLO exchange. During the HELLO exchange, each LSR proposes a value and the lower of the two is used as the hold time.

Targeted LDP hello-adjacencies between potentially non-directly connected LSRs are supported using an extended discovery mechanism. In this case, targeted hello messages are periodically sent to a specific IP address.

After the hello-adjacency is formed, the active LSR initiates establishment of a TCP connection to the peer LSR. At this point, an LDP session is initiated over the TCP connection. The LDP session consists of an exchange of LDP messages that are used to set up, maintain, and release the session.

Advertising Labels

You can control whether label mappings are advertised for:

- Direct routes
- *RIP* routes
- Static routes

In these cases, the switch is acting as the egress LER for these LSPs.

Propagating Labels

LDP propagates label mappings for FECs that exactly match a routing table entry.

In the case of label mappings received from an LDP peer, LDP checks for an exactly matching entry with a next hop IP address that is associated with the LDP peer from which a label mapping was received.

Label Advertisement Modes

LDP provides two modes for advertising labels:

- Downstream-on-demand (DoD)
- Downstream unsolicited (DU)

Using DoD mode, label bindings are only distributed in response to explicit requests. A typical LSP establishment flow begins when the ingress LER originates a label request message to request a label binding for a particular FEC (for a particular IP address prefix or IP host address). The label request message follows the normal routed path to the FEC. The egress LER responds with a label mapping message that includes a label binding for the FEC. The label mapping message then follows the routed path back to the ingress LSR, and a label binding is provided by each LSR along the path. LSP establishment is complete when the ingress LER receives the label mapping message.

Conversely, using DU mode, an LSR may distribute label bindings to LSRs that have not specifically requested them. These bindings are distributed using the label mapping message, as in downstream-on-demand mode. From an LDP message perspective, the primary difference using DU mode is the lack of a preceding label request message.

Architecturally, the difference is more significant, because the DU mode is often associated with a topology-driven strategy, where labels are routinely assigned to entries as they are inserted into the routing database. In either case, an LSR only uses a label binding to switch traffic if the binding was received from the current next hop for the associated FEC.

Both label advertisement modes can be concurrently deployed in the same network. However, for a given adjacency, the two LSRs must agree on the discipline. Negotiation procedures specify that DU mode be used when a conflict exists when using Ethernet links. Label request messages can still be used when *MPLS* is operating in unsolicited mode.

The Extreme LDP implementation supports DU mode only.

Label Retention Modes

LDP provides two modes for label retention:

- Conservative
- Liberal

Using conservative label retention mode, an LSR retains only the label-to-FEC mappings that it currently needs (mappings received from the current next hop for the FEC).

Using liberal retention mode, LSRs keep all the mappings that have been advertised to them. The trade-off is memory resources saved by conservative mode versus the potential of quicker response to routing changes made possible by liberal retention (for example, when the label binding for a new next hop is already resident in memory).

The Extreme *MPLS* implementation supports liberal label retention only.

LSP Control Modes

LDP provides two LSP control modes:

- Independent
- Ordered

Using independent LSP control, each LSR makes independent decisions to bind labels to FECs. By contrast, using ordered LSP control, the initial label for an LSP is always assigned by the egress LSR for the associated FEC (either in response to a label request message or by virtue of sending an unsolicited label mapping message).

More specifically, using ordered LSP control, an LSR only binds a label to a particular FEC if it is the egress LSR for the FEC, or if it has already received a label binding for the FEC from its next hop for the FEC. True to its name, the mode provides a more controlled environment that yields benefits such as preventing loops and ensuring use of consistent FECs throughout the network.

The Extreme *MPLS* implementation supports ordered LSP control only.

MPLS Routing

This section describes how *MPLS* and IP routing work together to forward information on your network.

MPLS provides a great deal of flexibility for routing packets. Received IP unicast frames can be routed normally or tunneled through LSPs. If a matching FEC exists for a received packet, the packet may be transmitted using an LSP that is associated with the FEC. The packet is encapsulated using an MPLS shim header before being transmitted.

Received MPLS packets can be label switched or routed normally toward the destination. Packets that are in the middle of an LSP are label switched. The incoming label is swapped for a new outgoing label and the packet is transmitted to the next LSR. For packets that have arrived at the end of an LSP (the egress end of the LSP), the label is popped. If this label is the bottom of the stack, the shim header is stripped and the packets are routed to the destination as normal IP packets.

Note
Multicast routing and *ECMP (Equal Cost Multi Paths)* on MPLS are not supported.

An MPLS domain is generally defined to be an *OSPF* or IS-IS autonomous system (AS). You can use MPLS to reach destinations inside one of these AS types. You can also use MPLS to tunnel through all or part of an AS in order to reach destinations outside of the AS.

MPLS Layer Details

MPLS can be thought of as a shim-layer between Layer 2 and Layer 3 of the protocol stack. MPLS provides connection services to Layer 3 functions while making use of link-layer services from Layer 2. To achieve this, MPLS defines a shim header that is

inserted between the link layer header and the network layer header of transmitted frames. The format of a 32-bit MPLS shim header is illustrated in the following figure.

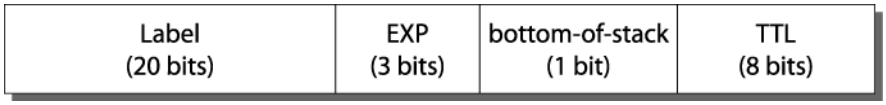


Figure 119: MPLS Shim Header

MPLS Shim Header

The *MPLS* shim header contains the following fields:

- 20-bit label
- 3-bit experimental (EXP) field

  The EXP field can be used to identify different traffic classes to support the DiffServ *QoS (Quality of Service)* model.
- 1-bit bottom-of-stack flag

  The bottom-of-stack bit is set to 1 to indicate the last stack entry.
- 8-bit Time-To-Live (TTL) field.

  The TTL field is used for loop mitigation, similar to the TTL field carried in IP headers.

MPLS Label Stack

The format of an *MPLS* label stack containing two MPLS shim header entries is shown in Figure 120.

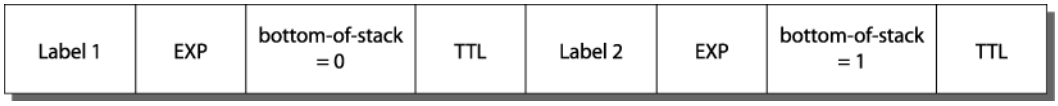


Figure 120: MPLS Label Stack

Figure 121 illustrates the format of a unicast MPLS frame on an Ethernet link. The MAC addresses are those of the adjacent MPLS router interfaces. The x8847 Ethertype value indicates that the frame contains an MPLS unicast packet. A different Ethertype value (x8848) is used to identify MPLS multicast packets.

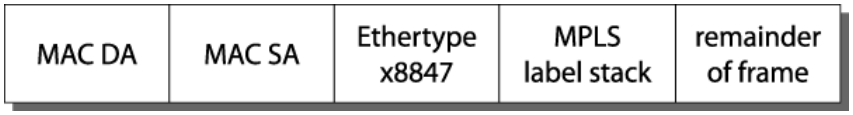


Figure 121: MPLS Unicast Frame on Ethernet

Figure 122 shows the format of a unicast MPLS frame that contains an 802.1Q *VLAN* tag. In both cases, the Ethertype values no longer identify the network layer protocol type. This implies that, generally, the protocol type must be inferable from the MPLS label value(s). For example, when only one type of protocol is carried on a given LSP.

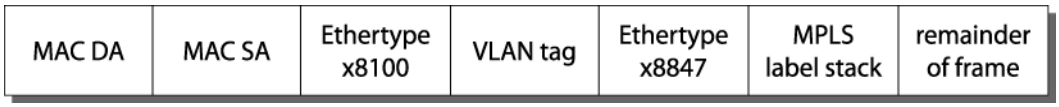


Figure 122: MPLS Unicast Frame on Tagged Ethernet VLAN

Note
For more detailed information on MPLS encapsulations, see RFC 3032, MPLS Label Stack Encoding.

Penultimate Hop Popping

Penultimate hop popping (PHP) is an LSR label stack processing optimization feature. When enabled, the LSR can pop (or discard) the remaining label stack and forward the packet to the last router along the LSP as a normal Ethernet packet.

By popping the label stack one hop prior to the LSP egress router, the egress router is spared having to do two lookups. After the label stack has been popped by the penultimate hop LSR, the LSP egress router must only perform an address lookup to forward the packet to the destination.

PHP label advertisements using implicit NULL labels can be optionally enabled. Support for receiving implicit NULL label advertisements by neighbor LSRs is always enabled. For example, if an LSR advertises implicit NULL labels for IP prefixes, the neighbor LSRs must support PHP.

Label Binding

Label binding is the process of, and the rules used to, associate labels with FECs.

LSRs construct label mappings and forwarding tables that comprise two types of labels: labels that are locally assigned and labels that are remotely assigned.

Locally assigned labels are labels that are chosen and assigned locally by the LSR. For example, when the LSR assigns a label for an advertised direct interface. This binding information is communicated to neighboring LSRs. Neighbor LSRs view this binding information as remotely assigned.

Remotely assigned labels are labels that are assigned based on binding information received from another LSR.

Label Space Partitioning

The Extreme implementation partitions its label space as described in the following illustration.

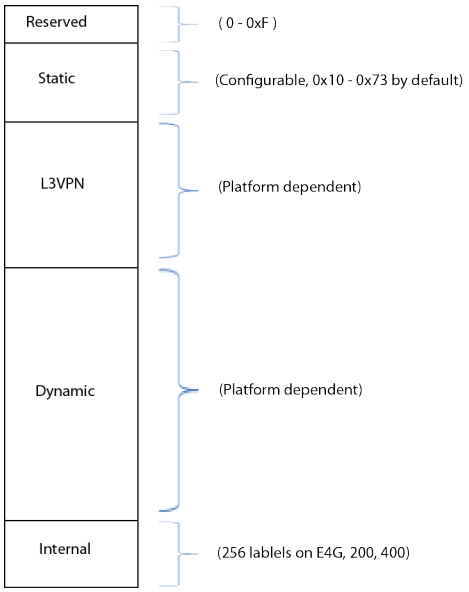


Figure 123: MPLS Label Space Partitions

Platforms that support *MPLS* divide the incoming (MPLS egress) label space into static, L3VPN, and dynamic label ranges. The static label range starts immediately after the reserved labels at 16 (0x10), and spans up through the configured maximum number of static labels. In ExtremeXOS 15.4, the `configure mpls label max-static` command is added to allow the static partition range to be changed. Any changes made to the static range affect subsequent ranges.

VRFs used for L3VPNs are allocated from a label range that immediately follows the static range. The remaining label space is called the dynamic label space, and is used by LDP and RSVP-TE. The `show mpls label usage` command displays the label ranges on a given system. The outgoing (MPLS ingress) label space spans the full 20-bit label range, and is used for all outgoing labels for both LSPs and PWs, static and signaled.

No hard limits are imposed on the maximum size of the label stack, other than the constraint of not exceeding the maximum frame size supported by the physical links comprising the LSP.

Jumbo frames should be enabled on all ports, and the jumbo frame size should be set to accommodate the addition of a maximally-sized label stack. For example, a jumbo frame size of at least 1530 bytes is needed to support a two-level label stack on a tagged Ethernet port, and a jumbo frame size of at least 1548 bytes is needed to support a VPLS encapsulated MPLS frame.

Routing Using Matching and Calculated LSP Next Hops

Normally, a route table prefix is associated with a gateway or next hop IP address.

Using *MPLS*, a route table prefix can also be associated with an LSP that can be used as the next hop. There are two types of LSP next hops that can be used to route a packet to its destination:

- Matching LSP next hop

An LSP is considered matching with respect to an FEC if it has been associated with that FEC via LDP or RSVP-TE. An example of this is an IPv4 prefix for which a matching label mapping has been received by LDP. Matching LSPs are supported for all route origin types.

- Calculated LSP next hop

  An LSP is considered calculated with respect to an FEC if it has been associated with that FEC through a routing protocol. Both *OSPF* and *BGP* can perform the calculations necessary to associate a route table prefix with an LSP next hop.

Figure 124 illustrates the concept of matching and calculated LSPs.

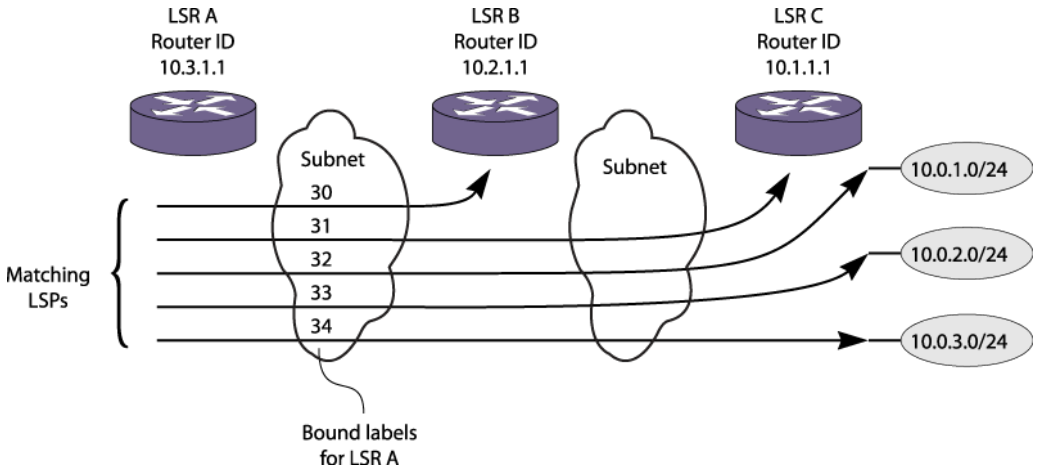


Figure 124: Matching and Calculated LSP Next Hops

Table 75 describes the label bindings in the MPLS forwarding table for LSR A that are maintained for FECs reachable via LSR A to LSR C, shown in Figure 124.

Table 75: Label Bindings for LSR A

Destination	Next Hop	Matching LSP Next Hop Label	Calculated LSP Next Hop Label
10.1.1.1/32	10.2.1.1	31	30
10.0.1.0/24	10.2.1.1	32	31
10.0.2.0/24	10.2.1.1	33	31
10.0.3.0/24	10.2.1.1	34	31

Matching LSP Next Hops

A matching LSP next hop is always preferred over a calculated LSP next hop.

Matching LSP next hop entries are added to the route table when an LSP becomes operational. They are deleted when an LSP is no longer operational.

OSPF Calculated LSP Next Hops

Managing calculated LSP next hop entries is more involved.

The *OSPF* Shortest Path First (SPF) algorithm checks the availability of LSPs to remote OSPF routers during a calculation. The intra-area SPF algorithm begins with the calculating router as the root of a graph. The graph is expanded by examining the networks connected to the root and then examining the routers connected to those networks. Continuing in this manner, the graph is built as a series of parent and child nodes. A check is made for a matching LSP next hop as each entry is added. A check is also made for an LSP next hop that can be inherited from the parent node. These inherited LSP next hops are referred to as calculated LSP next hops. Thus, for each route table entry, the modified SPF algorithm determines whether a matching LSP next hop is available and whether a calculated LSP next hop is available for use whenever a matching LSP next hop is not present.

The modification to the SPF algorithm described above is important, because it enables the capabilities provided by LDP or RVSP-TE LSPs to be fully utilized, while minimizing the resources devoted to label management.

For example, in a network where all the LERs/LSRs implement this feature (such as an all-Extreme *MPLS* network), labels only need to be advertised for the router IDs of the LERs/LSRs. Yet, LSPs can still be used to route traffic destined for any OSPF route.

More specifically, LSPs can be used for all routes advertised by OSPF, with the possible exception of LDP LSPs to routes summarized by OSPF area border routers (ABRs). The problem with using routes summarized by OSPF ABRs is that route summarization can prevent label mappings from being propagated for the links internal to the area being summarized, since an LSR only propagates LDP labels for FECs that exactly match a routing table entry.

BGP Calculated LSP Next Hops

BGP can also calculate how to use LSPs to reach BGP next hops.

For example, an IBGP session is established across the *OSPF*/*MPLS* backbone, and the communicating routers run both OSPF and IBGP. When an IBGP route is installed, BGP determines whether a matching LSP next hop exists to the destination. If not, it checks for an LSP to the BGP next hop. If an LSP exists to the BGP next hop, that LSP is used as an LSP next hop for the IBGP route.

The recalculation requirements for BGP are similar to those for OSPF; when an LSP to a BGP next hop router changes state, the BGP routing table entries must be checked to ensure their LSP next hop information is still valid.

LSP Precedence and Interaction

A longest prefix match (LPM) is determined for all packets.

If an LSP next hop is available, routed IP traffic may be forwarded over an LSP using the LSP next hop. With respect to a given prefix, LSP next hops can be either matching or calculated, and can be based on LDP, RSVP-TE, or static LSPs. Matching LSP next hops are preferred over calculated LSP next hops. RSVP-TE LSPs are preferred over LDP LSPs, and LDP LSPs are preferred over static LSPs. Also, RSVP-TE LSPs and static LSPs can be individually configured to enable or disable their use as LSP next hops.

Therefore, if a more preferred LSP is established, routed IP traffic may begin to use a new LSP next hop. Likewise, if a preferred LSP is torn down, routed traffic may begin to use the next best LSP next hop. These changes can take place when there is an *OSPF* routing topology change, an LDP label advertisement event, or a RSVP-TE signaling action.

Multivendor Support for Calculated LSPs

Unfortunately, some *MPLS* implementations do not support the ability to forward packets received on an egress LSP to their *OSPF* router ID and/or *BGP* next hop address.

If your MPLS network includes equipment that does not support this type of IP forwarding, you can use configuration commands to explicitly control the use of calculated LSP next hops.

The following commands enable and disable all use of LSP next hops. No IP traffic is routed over an LSP when mpls-next-hop IP routing capability is disabled.

- `enable iproute mpls-next-hop`
- `disable iproute mpls-next-hop`

These commands enable and disable the calculation of LSP next hops by OSPF:

- `enable ospf mpls-next-hop {vr vrf_name}`
- `disable ospf mpls-next-hop {vr vrf_name}`

These commands enable and disable the calculation of LSP next hops by BGP:

- `enable bgp mpls-next-hop`
- `disable bgp mpls-next-hop`

Layer 2 VPN over MPLS Overview (VPLS and VPWS)

Layer 2 virtual private networking (VPN) services over *MPLS* include Virtual Private LAN Services (VPLS) and Virtual Private Wire Services (VPWS).

These services enable Layer 2 VPN service offerings in a simple manner that is easy to deploy and operate. Layer-2 VPN services, based on a combination of Ethernet and MPLS/IP technologies, are designed to enable service providers to offer Business Ethernet private line services. These services use a simple Layer 2 interface at the customer edge and benefit from the resilience and scalability of an MPLS/IP core.

VPLS provides a virtual LAN between multiple locations. VPWS provides a virtual dedicated line between only two locations.

Layer 2 VPN Support

The LDP Layer 2 VPN implementation includes support for:

- LDP signaling support for pseudowire ID (PWid) FEC for pseudowire (PW) establishment.
- Use of LDP, RSVP-TE, or Static to establish transport LSPs.

- Tunnel endpoints, identified via configured IP addresses.
- Different *VLAN* IDs at each end of a PW, with the VLAN ID set by the egress switch to match that of the locally configured VLAN.
- Operations as VPLS, H-VPLS, or VPWS node.
- VLAN, VMAN, and port edge services (no port-qualified VLAN service).
- Flooding of Layer 2 packets to multiple PWs when operating as a VPLS or H-VPLS node.

Note
The implementation does not include support for pseudowire participation in running the Spanning Tree Protocol.

Layer 2 VPN Service Deliminators

Service deliminators are used to define how the customer is identified to the Layer 2 VPN service.

There are multiple types of service deliminators. The ExtremeXOS software currently supports three types. The first is a *VLAN* service. This service transparently interconnects two or more VLAN segments together over an *MPLS* network. The configured VLAN IDs for the customer switch interfaces are not required to match, as long as the egress LSR overwrites the VLAN tag with the locally defined VLAN ID.

The second service is a VMAN service, which interconnects two or more VMAN segments. The service operates like the VLAN service but uses the VMAN tag instead of the VLAN tag to identify the service. As with the VLAN service, the interconnected VMAN segments do not need to have matching VMAN IDs.

The third service is a port service, which transparently interconnects two or more ports together over an MPLS network. Traffic is transported unmodified between ports.

The VLAN and VMAN services are configured by adding the service VLAN or a VMAN to a VPLS or VPWS. The port service is not explicitly configured but is emulated using a combination of Layer 2 VPN capabilities. First a VMAN must be configured and the port added untagged to the VMAN. The service VMAN is then added to the VPLS or VPWS. At this point all traffic received on the port is VMAN encapsulated for transmission across the Layer 2 VPN. To transmit the traffic across the Layer 2 VPN as it was received on the port, the VPLS or VPWS is configured to exclude the service tag. By excluding the service tag, the VMAN tag is stripped prior to being transmitted from the switch. This configuration provides port mode service and allows one or multiple ports to be associated with a Layer 2 VPN.

MPLS Pseudowires

MPLS pseudowire (PW) tunnels are logical connections between two LERs over an LSP.

LDP Pseudowires are signaled based on the configured PW identifier (pwid). The signaled PW label is used to create a two-label-stack shim header on PW encapsulated packets. The outer label is the transport LSP label obtained from LDP or RSVP-TE and the inner label is the signaled PW label. LERs also signal the PW type when attempting

to establish a PW. The ExtremeXOS software supports only the PWid type FEC. The Generalized ID FEC type is currently not supported.

Note
MPLS PWs can also be configured with statically assigned labels.

Transporting 802.1Q Tagged Frames

When an 802.1Q Ethernet frame is encapsulated for transport over a VC tunnel, the entire frame is included, except for the preamble and FCS.

There is a configuration option that determines whether the 4-byte *VLAN* tag field is included in the transmitted packet. By default, the tag field is not included. If the tag field is not included, the egress LER may add one. If it is included, the tag service identifier may be overwritten by the egress LER. The ability to add a tag field or to overwrite the service identifier at the egress node allows two (possibly independently administered) VLAN segments with different VLAN IDs to be treated as a single VLAN.

The following command can be used to include the VLAN tag field:

```
configure l2vpn [vpls vpls_name | vpws vpws_name] {dot1q [ethertype
hex_number | tag [include | exclude]]} {mtu number}
```

This command can also be used to control the overwriting of the 802.1Q ethertype when the VLAN tag field is included. In this case, the ingress node prior to transmitting the encapsulated packet overwrites the ethertype. This allows devices with a configurable VLAN tag ethertype to interoperate.

Establishing LDP LSPs to PW Endpoints

Establishing a PW requires both an LSP and a targetted LDP session between the two endpoints.

The local PW endpoint is the *MPLS* LSR ID. The remote PW endpoint is identified using an IP address configuration parameter.

When using LDP to establish the LSPs, each endpoint needs to advertise a label mapping for an LSP to its local endpoint address. To ensure that its LDP peers use the label mapping, a corresponding IGP route should also be advertised for the address. The IGP route can come from any of the supported routing protocols, such as *OSPF*, or IS-IS. For example, when using OSPF, an OSPF route with prefix length 32 should be advertised for the configured IP address.

We recommend that you configure a loopback *VLAN* using the IP address of the local endpoint (the MPLS LSR ID). Use prefix length 32 for the IP address configured for the loopback VLAN. When you configure a loopback VLAN, the IP address used to identify the endpoint remains active, even when one or more of the LSR VLAN interfaces go down. Should a remote peer normally use one of the down interfaces, the normal IGP and LDP recovery procedures allow the PW to use one of the remaining up interfaces to minimize the network outage.

You should also configure the loopback VLAN for MPLS using the `configure mpls add vlan vlan_name` command. The addition of the loopback VLAN to MPLS causes LDP to include the IP address in LDP address messages. Some implementations (including the ExtremeXOS software) require this information to determine the correct LDP session over which to advertise label mappings for VC FECs (see Using LDP to Signal PW Label Mappings on page 1071).

Note
Neither MPLS nor LDP have to be enabled on the loopback VLAN.

There are two options to initiate the LDP advertisement of an LSP to the local MPLS LSR ID when a loopback VLAN has been configured for that IP address:

- Configure MPLS LDP to advertise a direct interface whose IP address matches the LSR ID and has prefix length 32. Use the `configure mpls ldp advertise direct lsr-id` command to do this.
- Configure MPLS LDP to advertise direct interfaces using the `configure mpls ldp advertise direct all` command.

Note
This causes LDP to advertise label mappings for all VLANs that have an IP address configured and have IP forwarding enabled.

While both of the above methods initiate the advertisement of a label mapping for an LSP to the local endpoint, the first method is the preferred method.

Using LDP to Signal PW Label Mappings

Just as LDP advertises label mappings for LSPs, it can also advertise label mappings for Layer 2 VPNs.

In this case, the signaled FEC information describes a particular Layer 2 VPN. This FEC is often called a Virtual Circuit FEC, or VC FEC. The VC FEC information includes a PWid that is a 32-bit numeric field. Unlike LSP label advertisements that are usually sent to all possible upstream peers, the VC FEC information is sent only to the configured remote endpoint.

When the first Layer 2 VPN is configured to a remote peer, *MPLS* automatically creates a targeted hello adjacency entity for establishing an LDP session. Once the session is established, LDP passes the VC FEC label mapping associated with the Layer 2 VPN. Once VC FECs for the same PW ID have been exchanged in each direction, MPLS is ready to associate the PW with an LSP to the remote endpoint as described in Message Types on page 1090.

To determine the correct LDP session over which to send a VC FEC, MPLS checks the IP addresses learned from its LDP peers via LDP address messages. The ExtremeXOS software MPLS expects to find the IP address of a remote PW peer among the addresses received over the LDP session to that peer.

To ensure that the local endpoint IP address is included in LDP address messages, it is highly recommended to configure MPLS on a loopback *VLAN* as described in Establishing LDP LSPs to PW Endpoints on page 1070.

Use the command `configure mpls add vlan vlan_name` to configure MPLS on the loopback VLAN. It is not required that LDP or MPLS be enabled on the VLAN for the associated IP address to be advertised in LDP address messages.

LSP Selection

A PW can be configured to use any available LSP to the peer endpoint IP address, or the PW can be configured to use one or more specific named LSPs.

In either case, the LSP has to egress (terminate) at the remote endpoint. In the case of an LDP LSP, the LSP's FEC has to be a /32 prefix length to the endpoint IP address. In the case of an RSVP-TE LSP or static LSP, the destination address has to be that of the remote endpoint. When configured to use any available LSP, *MPLS* gives preference to RSVP-TE LSPs, then to LDP LSPs, and lastly, to static LSPs. As a single LSP is chosen to carry the PW traffic, if multiple LSPs of the chosen type exist, the decision of which LSP of this type to use is non-deterministic.

The `configure l2vpn [vpls vpls_name | vpws vpws_name] peer ipaddress [add | delete] mpls lsp lsp_name` command forces the PW to use the specified named LSP. If multiple named LSPs are configured, only one is used to carry the PW. The decision of which of the multiple configured LSPs to use is non-deterministic.

RSVP-TE can be configured to allow specific types of traffic on an LSP. By default, LSPs are used to transport all traffic. Optionally, named LSPs can be configured to allow only IP traffic or only VPN traffic. This can be used to control the LSP selection for specific types of packets. For example, if both LDP and RSVP-TE LSPs exist and the RSVP-TE LSPs are configured to transport only VPN traffic, all IP traffic is forwarded using LDP LSPs. Since RSVP-TE LSPs are preferred over LDP LSPs, VPN traffic flows over the RSVP-TE LSPs. The following command configures this behavior for the specified RSVP-TE LSP:

```
configure mpls rsvp-te lsp lasp_name transport ip-traffic deny
```

Layer 2 VPN Domains

Layer 2 VPN domains are created by adding PWs to each peer LSR to build a fully-meshed interconnected VPLS.

For each peer added, a PW is signaled that is used to carry traffic from the local LSR to the remote peer LSR. Flood traffic from the local service (broadcast, multicast, and unknown unicast packets) is replicated and forwarded across all PWs in the VPLS. Each peer receives one copy of the packet for delivery to its locally attached service. As MAC learning occurs on PWs, unicast packets to a known destination MAC address are forwarded to the peer over the PW from which the MAC address was learned.

MAC Learning

Learned MAC addresses are associated with the PWs from which the packets are received.

The learned MAC address is always inserted into the *FDB (forwarding database)* as though it was learned on the local service *VLAN* (and not the VLAN identified in the dot1q tag in the received PW packet). MAC addresses learned from PWs use a different FDB aging timer than those MAC addresses learned on Ethernet interfaces. Different FDB aging timers are maintained for Ethernet and pseudowire Layer 2 VPN FDB entries. By default, both aging timers are set to 300 seconds. However, the aging timers for each type of FDB entry can be configured to different values. Note that PW FDB entries are not refreshed; they age out based on the configured aging timer setting, unless you have disabled the aging timer. Ethernet FDB entries automatically refresh with use, and do not age out unless they are not used for the length of time configured for the aging timer. Any MAC address associated with a PW is automatically cleared from the FDB when the PW label is withdrawn.

Note
MAC learning is disabled for VPWS.

Spanning Tree Protocols

There is some debate as to the benefit of supporting *STP (Spanning Tree Protocol)* within a Layer 2 VPN.

The idea is that STP protocols can be used to provide redundant VPN data paths that can be unblocked if the STP detects a spanning tree topology failure. In general, it is believed that introducing STP to VPLS increases network complexity with very little real benefit.

MPLS already provides a significant level of redundancy for the LSP over which a PW is carried. For example, if a PW is using an LDP established LSP, provided there are parallel routed paths to the PW endpoint, the PW automatically shifts from a withdrawn or failed LSP to the next best available LSP. For transport LSPs established using RSVP-TE, secondary LSPs can be configured that can be hot-swapped in the event of a primary LSP failure. Fast-reroute detour LSPs can also be used to protect RSVP-TE LSPs. Thus, even though the underlying transport LSP might have changed, the Layer 2 VPN data plane remains unaffected.

For these reasons, VPLS and STP are not normally enabled on the same *VLAN*. The exception is for local customer network redundancy such as shown in VPLS STP Redundancy Overview on page 1083.

When STP is not enabled on a VPLS VLAN, the BPDU functional address is not inserted into the *FDB* for this VLAN and all received BPDU packets are flooded across the Layer 2 VPN. In this scenario, a single large spanning tree topology spanning all interconnected Layer 2 VPN service sites is constructed. Note that this is not a recommended configuration for a Layer 2 VPN service. Depending on the packet latency within the backbone network, STP timers might need to be tuned to build and maintain reliable topologies.

Currently, most ExtremeXOS software Layer 2 protocols cannot be configured on MPLS Layer 2 VPN domains. Likewise, the following protocols cannot be enabled on a Layer 2 VPN service VLAN:

- *VRRP (Virtual Router Redundancy Protocol)*

- *ESRP (Extreme Standby Router Protocol)*
- *EAPS (Extreme Automatic Protection Switching)* control VLAN

IP Protocol Considerations

The ExtremeXOS software allows an IP address to be configured for a Layer 2 VPN service *VLAN*. This is permitted to allow the switch to use the IP ping and traceroute functions to and from other nodes on the VLAN. It is envisioned that any such IP address is configured temporarily to assist in network verification.

As such, the ExtremeXOS software does not allow IP forwarding to be enabled on a Layer 2 VPN service VLAN for either IPv4 or IPv6. Therefore, any higher-level protocol that requires IP forwarding to be enabled cannot be enabled on a Layer 2 VPN service VLAN. For example, *OSPF* cannot be enabled on a VPLS service VLAN at the PE router. However, OSPF routers can be attached to the service vlan transparently at the customer interfaces. In addition, *IGMP (Internet Group Management Protocol)* snooping cannot be enabled on a VPLS service VLAN.

MPLS Layer 2 VPN Characteristics

Characteristics of a Layer 2 VPN include:

- Use of LDP or RSVP-TE to establish the underlying pseudowire transport LSPs.
- Pseudowire endpoints are identified through configured VPLS/VPWS peer IP addresses.
- Configuration of a Layer 2 VPN pseudowire ID doubles as VPN ID.
- Customer packet *VLAN* tags may be overwritten on egress from the pseudowire allowing local service VLANs or VMANs interconnected over the same Layer 2 VPN to have different IDs.
- Customer packet VLAN tags can be included or excluded from packets transmitted on the pseudowire.
- Customer packet VLAN tag ethertype value can be modified before packet is transmitted on the pseudowire, allowing local service VLANs or VMANs interconnected over the same VPLS to have different 802.1Q tag ethertype values.
- Support for full-mesh VPN architectures.
- Support for VLAN, VMAN, and port Layer 2 VPN services.
- Support for enabling and disabling VPLS and/or the VPLS service.
- Support for enabling and disabling VPWS and/or the VPWS service.

Layer 3 VPN over MPLS Overview

Layer 3 VPN services over *MPLS* are supported using *BGP*. For more information, see Layer 3 VPN Configuration Overview on page 2211.

H-VPLS Overview

VPLS requires a full mesh of pseudowires between all Provider Edge (PE) peers.

As *MPLS* is pushed to the edge of the network, this requirement presents a number of problems. One problem is the increased number of pseudowires required to service a large set of VPLS peers. In a full-mesh VPLS, pseudowires must be established between all VPLS peers across the core. Full-mesh networks do not scale well due to the number pseudowires that are required, which is p(p-1), where p is the number of peer devices in the network. Hierarchical VPLS (H-VPLS) networks can dramatically increase network scalability by eliminating the p^2 scaling problem.

In a hierarchical VPLS network, a spoke node (often a Multi-Tenant Unit—MTU) is only required to establish a pseudowire to a single core PE. Thus the number of pseudowires required in the provider's network is c(c-1) + s, where c is the number of core PE nodes and s is the number of spoke MTU edge devices. This is a significant reduction in the number of pseudowires that need to be established and maintained. For example, a 10-core PE network with 50 MTU devices per core PE requires almost 260,000 pseudowires using a full-mesh VPLS design. A hierarchical VPLS design requires only 590 pseudowires.

An example H-VPLS network is shown in Figure 125.

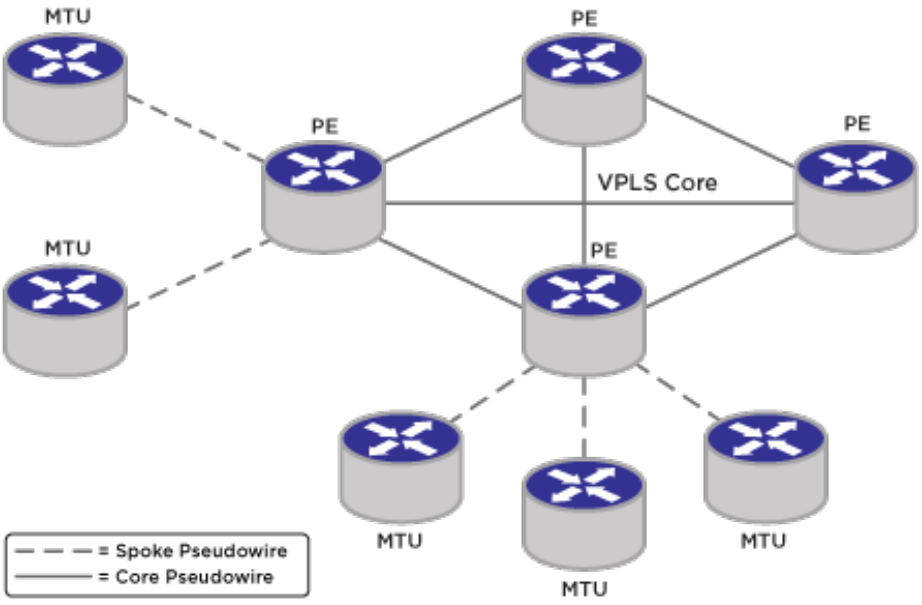


Figure 125: Example H-VPLS Network

H-VPLS spokes allow VPLS domains to be constructed hierarchically in a partial-mesh or hub-and-spoke configuration. This is useful for increasing the scaling of VPLS domains that can be supported. Within the context of H-VPLS, a spoke is a VPLS connection between two VPLS peers. Typically, one spoke node provides connectivity to the customer *VLAN* or customer service while its peer, a core node, provides repeater connectivity to other VPLS peers.

The pseudowire hierarchy must be known because the forwarding rules for spoke and core pseudowires are different. Flood traffic received on a core pseudowire from another full-mesh core PE must not be transmitted over other core pseudowires to other PEs. However, flood traffic received on a core pseudowire is transmitted on all

spoke pseudowires in the VPLS. Unlike core pseudowires in a full-mesh VPLS, flood traffic received on a spoke pseudowire must be transmitted on all other pseudowires in the VPLS, including pseudowires to other core PEs.

H-VPLS introduces the definition of a pseudowire type. In previous ExtremeXOS releases, only core peers were supported in an interconnected full-mesh configuration. Therefore, all pseudowires were considered to be of the type core. A new spoke pseudowire type is introduced and is highlighted in Figure 126 on page 1077. A VPLS core node that has multiple spoke pseudowires but no configured core pseudowires is informally referred to as a hub.

Eliminating Packet Replication by the MTU

A scaling problem inherent in a full-mesh VPLS network is packet replication.

In a full-mesh configuration, until a node learns over which pseudowire a MAC address is reachable, unknown unicast frames must be flooded on all pseudowires within the VPLS. Packet replication is always true for broadcast and multicast traffic. As the number of VPLS peers increase, the packet replication burden on a node increases. MTU devices attached to a full-mesh core most likely cannot maintain wire-speed forwarding as the number of VPLS peers increase. Hierarchical VPLS eliminates this MTU burden by requiring only a single pseudowire connection between a spoke and its core PE peer. Packet replication is pushed to the PEs, where it is more suitably handled.

Simplifying Customer Service Provisioning

Bandwidth provisioning between an MTU and a PE is extremely difficult with a full-mesh VPLS design.

Since each VPLS instance can require multiple tunnel LSPs, the bandwidth requirements for each tunnel LSP must be separately accepted and individually enforced by every PE a tunnel LSP traverses. Because the provider requirement is to manage the provisioned bandwidth for the VPLS and not each tunnel LSP, the MTU has the added responsibility of rate limiting the aggregate egress traffic across multiple tunnel LSPs on the uplink. Due to packet replication issues described previously, this is not practical.

Hierarchical VPLS designs simplify bandwidth provisioning and management. Because tunnel LSPs from the MTU are terminated at the PE, tunnel LSP resources are easily shared and managed between customers. Thus, traffic for multiple VPLS instances can be transported across a single tunnel LSP. In some cases only a single best-effort tunnel LSP is required between the MTU and the PE. Traffic for each customer is carried over a different pseudowire on the same tunnel LSP. This allows the tunnel LSP to be signaled once, with the desired bandwidth and priority parameters sufficient for providing best-effort service for customers connected to the spoke peer. If a customer upgrades their service or a new customer is connected that requires guaranteed bandwidth, a second tunnel LSP could be signaled with the SLA bandwidth parameters. Once established, the second tunnel LSP can carry traffic for a single customer as a premium service.

Redundant Spoke Pseudowire Connections

Redundant spoke pseudowires to PE peers can be configured from an MTU to provide backup connectivity into a VPLS core.

The addition of a redundant spoke pseudowire is optional. By default, when the MPLS Feature Pack license has been applied to the switch, the spoke pseudowire to the primary peer is used to forward packets. In the event of a network failure over the primary pseudowire, the spoke pseudowire to the secondary peer is used to provide redundant VPLS connectivity. An example network is shown in Figure 126.

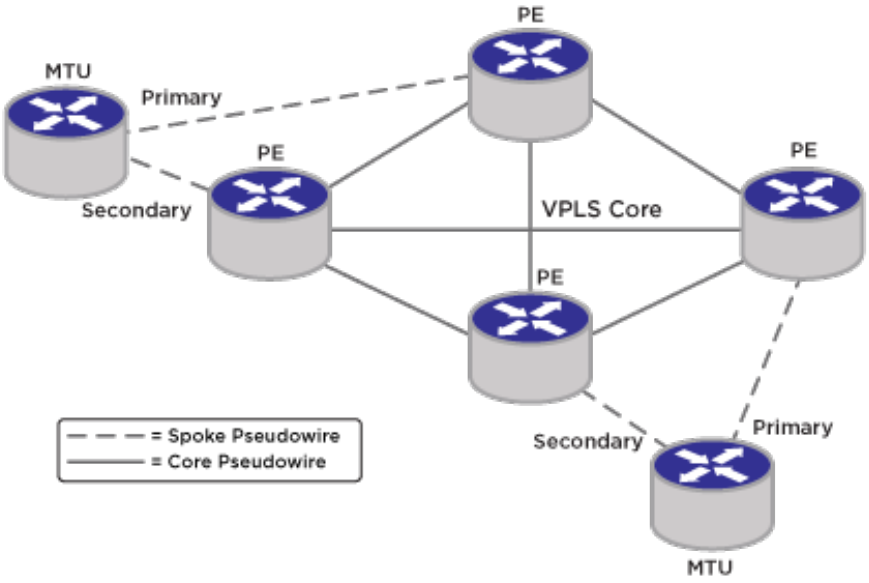


Figure 126: Example H-VPLS Network with Redundant Spokes

When both the primary and secondary pseudowires are established, the MTU is responsible for blocking the secondary pseudowire. Any packets received on the secondary pseudowire while the primary pseudowire is active are discarded. This behavior prevents packet-forwarding loops within the L2 VPN.

Since the MTU is responsible for choosing which pseudowire to the VPLS is active, the MTU is uniquely responsible for preventing network loops. The MTU uses only one spoke pseudowire per VPLS and only the label stack associated with the active pseudowire is programmed into the hardware. If the active pseudowire fails, then the label stack for the active pseudowire is removed from hardware. The secondary pseudowire label stack is then installed in the hardware in order to use the redundant VPLS link from the MTU into the VPLS core. Customer connectivity through the MTU should experience minimal disruption.

IETF RFC 6870 defines the "Preferential Forwarding" status bit to designate which pseudowire is active and can be used to forward user packets. The MTU sets this bit to indicate that a pseudowire is standby (not active) and should not be used to forward user packets.

When a failover occurs from a primary pseudowire to a secondary pseudowire, the MTU clears its *FDB* database of MAC addresses learned over the primary pseudowire. It then begins learning MAC addresses over the new active pseudowire. To inform other nodes to clear their learned MAC database, the MTU can send a MAC address-withdraw message (if this feature is enabled) to the peer PE node of the secondary pseudowire. This PE node can subsequently send its own MAC address-withdraw message to the other VPLS full-mesh core nodes. Upon receipt of the MAC address withdrawal message, each core node clears its database. In this manner, other core nodes re-learn MAC addresses from the correct pseudowire or port.

Packets can be received out-of-order by the VPLS destination device during certain pseudowire failover events. In the redundant VPLS spoke configuration, when the primary pseudowire fails, traffic is immediately switched to the secondary pseudowire. For a very short period of time, there may be packets that are in route via both pseudowires. No attempt to prevent mis-ordered packets from being received is made.

The command to configure the VPLS peer from an MTU to a PE and from a PE to an MTU is fundamentally the same. However, the optional primary and secondary pseudowire keywords are only applicable on the MTU since the MTU is responsible for preventing loops within the VPLS. A switch cannot be configured with a primary and a secondary pseudowire to the same peer within a VPLS. This is an invalid configuration since it provides no redundant protection for a failed PE node.

MAC Address Withdrawal TLV Support

MAC address withdrawal is a feature that is used to inform other nodes that certain FDB MAC address entries should be immediately unlearned, rather than waiting for them to age out. Traffic destined to these unlearned MAC addresses is then flooded until the MAC addresses are learned again. MAC address withdrawal is enabled by default when the MPLS Feature Pack License has been applied to the switch, but can be disabled. When this feature is disabled, traffic destined to MAC addresses previously learned over a now unusable pseudowire (PW) is not flooded until the FDB entries eventually age out. However, it can take the VPLS network longer to adjust than when MAC address withdrawal is enabled. This section describes how this feature operates when it is enabled.

After certain network recovery events, MAC addresses should be unlearned. For example, if the MTU's last usable transport LSP for the primary spoke pseudowire goes down, the MTU decides to activate and switch to the secondary pseudowire and send the primary and secondary core PE nodes an LDP notification message containing the appropriate PW status codes. The core PE node of the primary pseudowire flushes its FDB of any MAC addresses learned from the MTU, but does not send any MAC address-withdraw messages. It is the responsibility of the MTU to initiate the sending of MAC address-withdraw messages. If MAC address withdrawal is enabled, the MTU sends a MAC address-withdraw message to the core PE node of the secondary pseudowire. This node, in turn, sends its own MAC address-withdraw message to all the other core PE nodes in the VPLS causing each of these other core PE nodes to flush their FDB of any matching learned MAC addresses specified in the address-withdraw message. By withdrawing MAC addresses immediately, the other core PE nodes are forced to flood traffic destined to the MAC addresses specified in the address-withdraw message. If an

alternate VPLS path exists, the new path can be quickly learned without having to wait for the FDB MAC entry to age out.

When a node needs to withdraw a MAC address, it can signal the MAC withdrawal for a VPLS using an address-withdraw message in one of two ways: the MAC address is explicitly specified in a MAC TLV; or an empty MAC TLV is sent indicating that all MAC addresses not learned from that node should be unlearned. Because this information may be propagated to multiple VPLS nodes, a control plane processing trade-off exists. To reduce the processing load, ExtremeXOS sends an empty MAC TLV. Additionally, ExtremeXOS supports the processing of multiple withdraw messages per VPLS, since other vendors may choose not to send an empty MAC TLV.

Event Log Messages

MPLS has full EMS logging capabilities to capture error as well as debug information. Messages are logged with the MPLS component identifier.

SNMP Support

No *SNMP (Simple Network Management Protocol)* support is provided for H-VPLS.

Protected VPLS and H-VPLS with ESRP Redundancy Overview

Protected VPLS Access enables redundant nodes at the entry to a VPLS or H-VPLS network.

This feature provides fault tolerant connectivity from the customer *VLAN* to the backbone VPLS. This could be implemented by running Layer 2 protocols across the VPLS to block switch ports, but this can lead to sub optimal spanning tree topologies across the VPLS backbone and relatively long outages while the *STP* converges. Instead, the ExtremeXOS software has been enhanced to provide the ability to configure redundant VPLS switches using a dual-homed design that provides fast failover for protected access points.

The first figure below shows fault-tolerant access in a full mesh core VPLS network while the second figure shows fault-tolerant access in a hierarchical VPLS network.

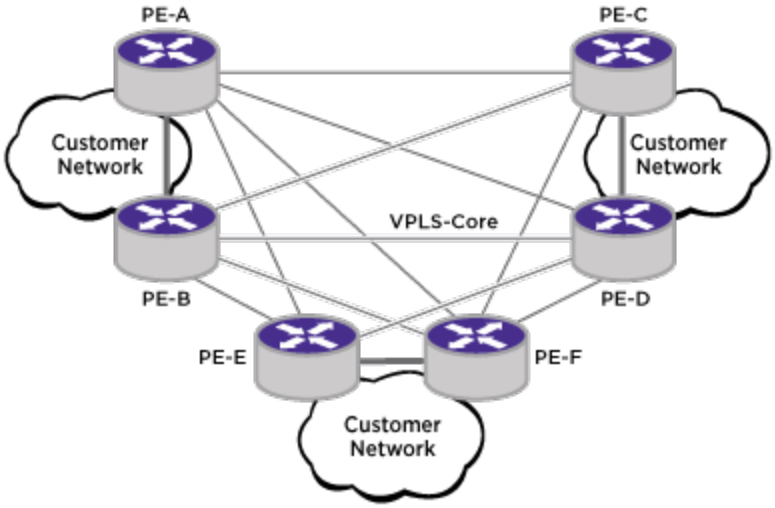


Figure 127: Example Protected Access VPLS Network

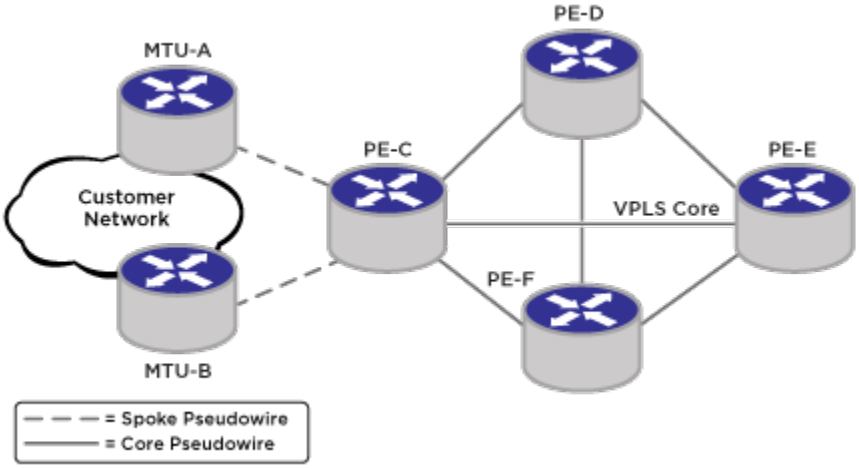


Figure 128: Example Protected Access H-VPLS Network

In the above figure, fault tolerance is provided at the customer site by MTU-A and MTU-B. A failure of either MTU-A or MTU-B does not result in any loss of customer connectivity beyond the failover time from one MTU to the other.

ESRP is employed to ensure that only one VPLS switch is active at any instant in time for a protected customer access point. Only the ESRP master switch forwards packets between the access network and the backbone VPLS network. This active primary switch retains this status based on a set of predefined tracking criteria. If the configured criteria can be better satisfied by the inactive secondary VPLS switch, the primary VPLS switch relinquishes control and the secondary switch assumes the active role. The secondary switch can also autonomously assume the active role if it detects that the primary switch has failed. This use of ESRP helps to prevent duplicate packet delivery and to prevent broadcast loops when the customer network is a loop topology.

Fault Tolerant Access Points Assumptions and Limitations

The following assumptions and limitations are associated with a fault tolerant access point network:

- This feature supports a maximum of two redundant VPLS switches per protected *VLAN* access point.
- This feature operates only with *ESRP* extended mode.
- For ESRP to communicate between neighbor switches, you must configure a separate control VLAN with the same network layout as the set of protected customer VLANs. For example, consider two customer VLANs, VLAN-X and VLAN-Y. Both require protected VPLS access. If both VLAN-X and VLAN-Y have the same network layout (for example, both are part of a single *EAPS* domain), you must create an ESRP control VLAN that has the same layout as VLAN-X and VLAN-Y. Conversely, if VLAN-X and VLAN-Y do not have the same layout, you must create two separate ESRP domains with each control VLAN following the layout of the associated service VLAN(s).
- All VPLS switches in the control VLAN need to have the same ESRP domain configured. VPLS switches that provide protected access to the VPLS network need to have ESRP enabled while other nodes in the control VLAN need to be ESRP aware.
- The software does not validate the configuration between switches to determine if all VPLS switches for a protected VLAN are configured to be part of the same ESRP.

H-VPLS Redundant Edge Network

In Figure 129, PE-C still represents a single point of failure. To remove this exposure, the fault tolerant access points can be combined with the redundant spoke pseudowires to produce a redundant edge configuration, as shown below.

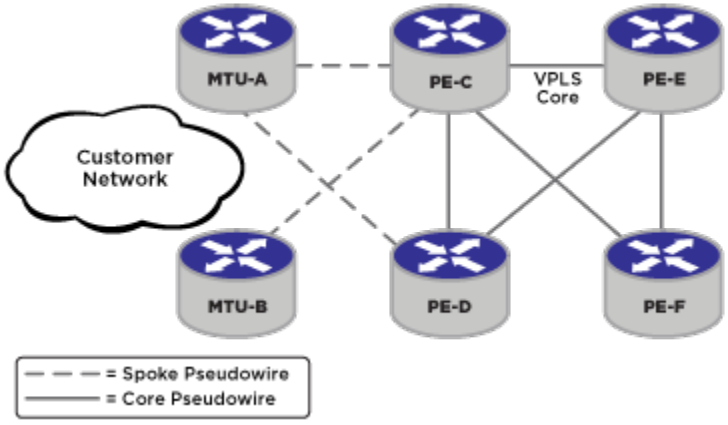


Figure 129: Example Redundant Edge H-VPLS Network

In the network shown in the above figure, only one of the pseudowires between the MTUs and their attaching PEs is active at any instant in time. This network provides fault tolerance for a failure at either MTU or at either of the attaching PEs, as well as for the active pseudowire in use between the MTUs and PEs.

Fault Tolerant VPLS Operation

To provide the fault tolerance shown in the following figure through the following figure, the redundant network nodes communicate with each other to determine the active primary and inactive secondary status and elect an active master node.

For a VPLS domain type, *ESRP* considers election factors in the following order: standby, active ports, tracking information, stickiness, ESRP priority, and MAC. For more information on the ESRP election priority, see ESRP.

For fault tolerant VPLS to function correctly, the ExtremeXOS software imposes restrictions on the configuration options on the VPLS redundancy type ESRP domain. Table 76 lists the configuration restrictions on the control *VLAN*.

Table 76: ESRP Configuration Restrictions for VPLS Type Domains

No.	Parameter	Restrictions	Remarks
1	*mode*	Not configurable	Only extended mode is supported.
2	*elrp poll*	Always enabled on control VLAN ports	Enabled because control VLANs can have loops. That is, a control VLAN is protecting an S-VLAN in an *EAPS* ring.
3	*master VLAN*	No restrictions	The ESRP control VLAN is configured as the master VLAN on the ESRP master and slave.
4	*member VLAN*	Not configurable	Member VLANs are not configured for the domain since we would need the slave node to perform L2 switching and L3 forwarding.
5	*track-environment*	Not configurable	Tracking is always done on pseudowires.
6	*track-VLAN*	Not configurable	Tracking is always done on pseudowires.
7	*track-IProute*	Not configurable	Tracking is always done on pseudowires.
8	*track-Ping*	Not configurable	Tracking is always done on pseudowires.
9	*domain-id*	None	
10	*elrp-master-poll*	No restrictions	Not configurable, disabled by default.
11	*elrp-premaster-poll*	No restrictions	Not configurable, disabled by default.
12	*election policy*	Not configurable	Always set to: standby > ports > track > sticky > priority > mac.
13	*name*	No restrictions	

Table 76: ESRP Configuration Restrictions for VPLS Type Domains (continued)

No.	Parameter	Restrictions	Remarks
14	*priority*	No restrictions	
15	*timer*	No restrictions	When the service VLAN is part of an EAPS ring, it is strongly recommended that the hello timer is always greater than the EAPS master health checkup timer because an ESRP switch before EAPS could cause traffic loops.

The MTU nodes in the following figure signal the PE nodes about active/inactive state using the Status TLV defined in RFC 4447, Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP). This operation is described in Redundant Spoke Pseudowire Connections on page 1077.

Performance of Fault Tolerant VPLS Access Points

Switching times for a VPLS type *ESRP* domain are the same as that for regular domains. Actual time depends on the configured hello timer values and is expected to be in the order of seconds.

Deployment and Application Considerations

The following configuration guidelines should be observed when deploying fault tolerant access points:

- When this feature is deployed with *EAPS* access rings, it is strongly recommended that the EAPS health check time is configured to a value that is less than or equal to the *ESRP* hello time. Current defaults are one second and two seconds for EAPS health check and ESRP hello, respectively.
- All nodes in the control *VLAN* other than the two VPLS enabled nodes are configured as ESRP-Aware.

Event Log Messages

MPLS has full EMS logging capabilities to capture error as well as debug information. Messages are logged with the MPLS component identifier.

SNMP Support

No *SNMP* support is provided for H-VPLS.

VPLS STP Redundancy Overview

Figure 130 shows an example network that uses *STP* to support redundant links to an H-VPLS network.

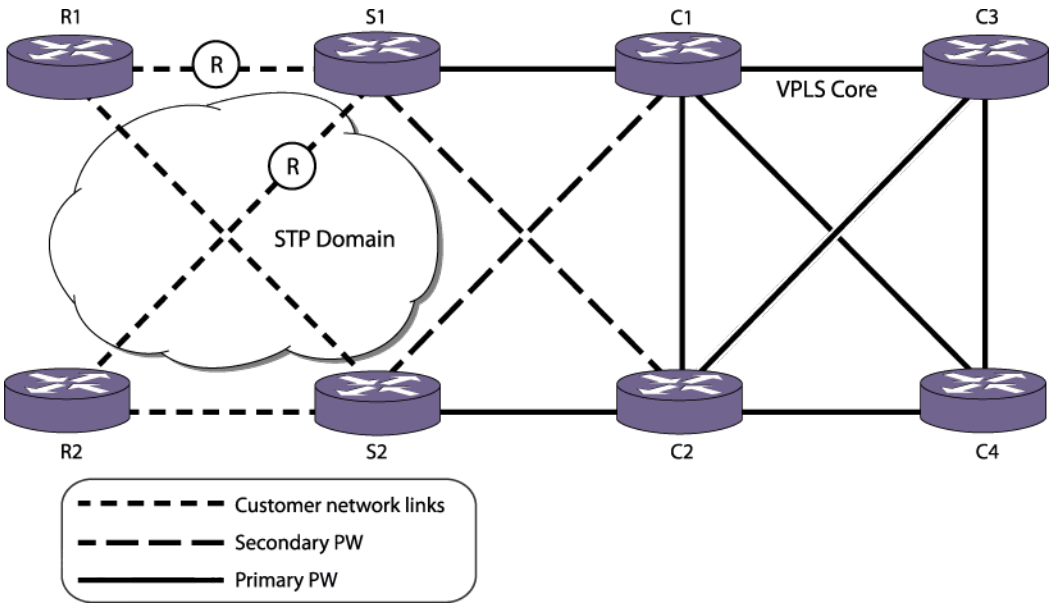


Figure 130: Redundant Edge H-VPLS Network with STP Example

This topology uses redundant VPLS spoke nodes (S1 and S2) and an STP customer network to protect customer access. The redundant VPLS nodes provide protection from the loss of a VPLS node, and STP provides protection from the loss of a node or link in the customer access network. Within the VPLS nodes, VPLS and STP work together to react to topology changes in the customer access *VLAN*.

This topology uses the restricted role feature on access switch ports to control path redundancy. In Figure 130, the VPLS nodes S1 and S2 are the lowest priority STP bridges (STP prefers lower priority for root bridge election and shortest path calculation). The S1 ports connected to the R links are configured for STP restricted role mode. To prevent network loops, the restricted role mode in S1 blocks an STP enabled port when STP BPDUs with better information are received from the access network. As shown, the customer traffic uses S2 to access the VPLS network. Should one of the two restricted ports on S1 become unblocked due to a topology change, customer traffic could use both S1 and S2 to access the VPLS network.

The selection of primary and secondary PWs for this configuration is arbitrary. Therefore data paths traversing the spoke nodes S1 or S2 could use either or both core nodes C1 and C2.

In the network shown in Figure 130, traffic destined to R1 from the VPLS core traverses C2, S2, and R1.

Failure Recovery Scenario without VPLS STP Redundancy

If VPLS *STP* redundancy is not configured on the network shown in the following figure and the S2-R1 link fails, S1 unblocks S1-R1 and S2 flushes its *FDB* on the S2-R1 port.

The VPLS core nodes C3 and C4 are unaware of this change and continue to forward any traffic destined for R1 to C2. C2 continues to forward traffic towards S2. This results in data loss because S2 is not able to reach R1 over the customer network.

This data loss continues until one of the following events occurs:

- The FDB entry for R1 ages out in the VPLS core nodes
- R1 sends traffic into the VPLS core allowing its MAC address to be relearned by the VPLS core nodes

Depending on the type of data traffic from R1, the latter scenario might not occur quickly.

Failure Recovery Scenario with VPLS STP Redundancy

If VPLS *STP* redundancy is configured on switches S1 and S2 in the following figure and the S2-R1 link fails, S1 unblocks S1-R1 and S2 flushes its *FDB* on the S2-R1 port.

When S2 flushes its FDB, it also sends a flush message to its core peer C2. Upon receipt of this flush message, C2 flushes its FDB entries learned over the PW to S2. C2 also sends flush messages to its core peers C1, C3, and C4. These core nodes then flush their FDB entries learned over their PWs to C2. Any traffic from the VPLS core destined for R1 is flooded until such time as traffic from R1 is forwarded into the VPLS core.

Note
In the example shown in Figure 130 on page 1084, the core nodes are unaware that STP redundancy is configured on S1 and S2. There is no STP redundancy configuration on C1 and C2.

The following figure shows an H-VPLS configuration where the STP network directly connects to redundant spoke nodes. A similar configuration is possible where the STP network directly connects to redundant VPLS core nodes. In this case, the core nodes participating in STP are configured for STP redundancy and originate flush messages to their core peers whenever STP causes a flush on an STP port.

Requirements and Limitations

The solution shown in VPLS EAPS Redundancy Overview on page 1086 has the following requirements and limitations:

- The redundant VPLS nodes must be core nodes.
- An *EAPS* common link is required between the redundant VPLS nodes.
- The redundant VPLS nodes (Core 1 and Core 2) use PWs to connect to each other and the other core nodes (Core 3 and Core 4).
- The redundant VPLS nodes use PWs to support the EAPS-protected VLANs, not the EAPS common link.
- Works only with EAPS customer attachment ring.
- The EAPS master should *not* be on a VPLS node.
- EAPS state used by VPLS to control state of PWs (Active/Ready).
- EAPS monitors common link state and ring state.
- VPLS on controller node uses EAPS state to set Active/Ready PWs.
- EAPS blocks customer-facing ports as normal.

Enabling and Disabling VPLS STP Redundancy

- To enable or disable VPLS *STP* redundancy, use the following commands:

  `configure {l2vpn} vpls vpls_name redundancy [esrp esrpDomain | eaps | stp]`

  `unconfigure {l2vpn} vpls vpls_name redundancy [eaps | esrp | stp]`

VPLS EAPS Redundancy Overview

To protect your customer access network from link and node failures, you can use the VPLS with Redundant *EAPS* configuration shown in Figure 131.

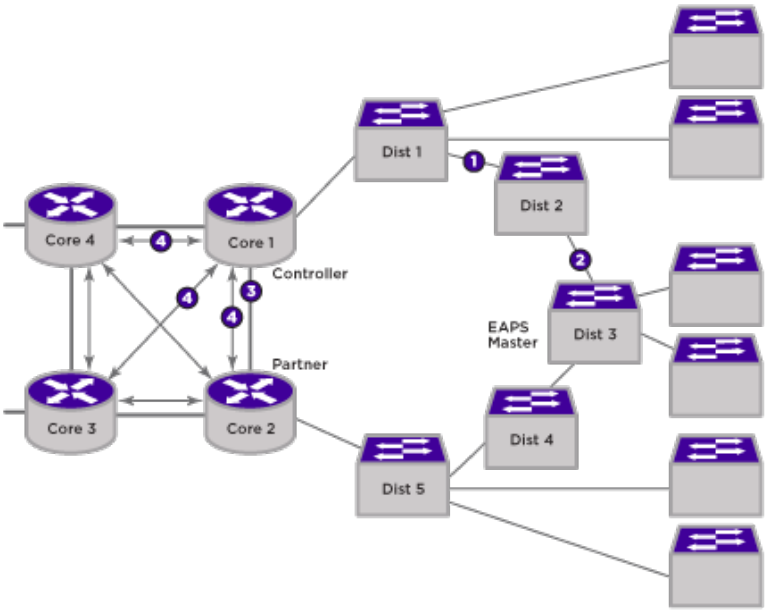


Figure 131: VPLS with Redundant EAPS Configuration Example

This topology uses redundant VPLS core nodes and an EAPS ring to protect customer access. The redundant VPLS nodes provide protection from the loss of a VPLS node, and the EAPS ring provides protection from the loss of a node or link on the EAPS ring. Within the VPLS nodes, VPLS and EAPS work together to control the use of PWs so that no loops are created.

During normal operation, the Dist2 node is the EAPS ring master, and it blocks the port leading to the link at point 2. The protected VLANs on the EAPS ring do not include the EAPS common link at point 3. The protected VLANS on the EAPS ring use VPLS PWs between the redundant VPLS nodes to connect the two ring segments. This difference in normal EAPS operation is established during configuration by configuring an EAPS-protected VLAN on the core node with only one port on the ring.

To see the commands used to create the configuration shown above, see VPLS with Redundant EAPS Configuration Example on page 1133.

Requirements and Limitations

The solution shown in VPLS EAPS Redundancy Overview on page 1086 has the following requirements and limitations:

- The redundant VPLS nodes must be core nodes.
- An *EAPS* common link is required between the redundant VPLS nodes.
- The redundant VPLS nodes (Core 1 and Core 2) use PWs to connect to each other and the other core nodes (Core 3 and Core 4).
- The redundant VPLS nodes use PWs to support the EAPS-protected VLANs, not the EAPS common link.
- Works only with EAPS customer attachment ring.
- The EAPS master should *not* be on a VPLS node.
- EAPS state used by VPLS to control state of PWs (Active/Ready).
- EAPS monitors common link state and ring state.
- VPLS on controller node uses EAPS state to set Active/Ready PWs.
- EAPS blocks customer-facing ports as normal.

Failure Recovery Scenario 1

Suppose that a failure occurs at point 1 in the following figure.

The *EAPS* master detects the topology change (either through a failure notification from a node on the ring or through a hello timeout), and it unblocks the port on the protected *VLAN* at point 2. The Dist 2 node now connects to the VPLS through Core 2 instead of through Core 1.

When the topology changes either on an access ring or the shared port link, the path used to reach customer devices can change. For example, in the following figure, the path that Dist 2 takes to reach other parts of the VPLS network changes following the failure on the access ring at point 1. Prior to the failure, Dist 2 used Core 1 to reach the VPLS network. Following the failure, Dist 2 accesses the VPLS network using Core 2.

When the EAPS master detects a topology change, it sends a flush *FDB* message to its transit nodes. The transit nodes re-learn all the MAC addresses on the ring. However, this flush FDB message is not propagated over the VPLS network. As a result, Core 3 in the following figure still expects to find Dist 2 through the PW between Core 3 and Core 1. Any traffic destined for Dist 2 that is sent to Core 1 will not reach its destination.

To correct this problem, EAPS informs VPLS about any received EAPS flush FDB messages on both the controller and the partner nodes, and VPLS performs a local flush of any MAC addresses learned from the originating nodes. In this example, the EAPS processes in both Core 1 and Core 2 notify VPLS because neither node knows where the access ring is broken. The VPLS services in Core 1 and Core 2 send flush messages to the other VPLS nodes.

Failure Recovery Scenario 2

The recovery scenario is more complicated when the common link fails between Core 1 and Core 2.

The common link is configured to be an *EAPS* common link, and this configuration requires that one node be designated as the EAPS controller node and the other node be designated as the EAPS partner node. As described below, the selection of which node is assigned which role must consider the overall customer topology.

If the shared port link fails, the EAPS master node unblocks its port, and VPLS on the EAPS controller node takes the additional action to remove the PWs (point 4 in the following figure) associated with the VPLS from hardware.

In this recovery mode, all traffic to and from the access ring and the rest of the VPLS instance passes through Core 2. When the EAPS controller node detects that the common link is repaired, it enters the preforwarding state as normal. When the controller node exits the preforwarding state, EAPS informs VPLS so that VPLS can reestablish the PWs.

Failure Recovery Scenario 3

Now suppose there is a failure on the shared port link (point 3) and on the access ring at point 1.

The recovery actions for this double-failure need to be somewhat different. In this case, even though the core link has failed, both core nodes do not receive a copy of the ring traffic. For example, the only path to the VPLS network for Dist 1 is through the controller core node. In this case, the controller node does not take down its PWs.

It is possible that the customer access network could have parallel *EAPS* rings that attach to Core 1 and Core 2 as shown in Figure 132. In this example, the network connections are broken at each point X and as long as any of the parallel EAPS rings are complete, there is a path to both core VPLS nodes. Thus, the controller node must take down its PWs as long as any of the parallel EAPS rings is complete.

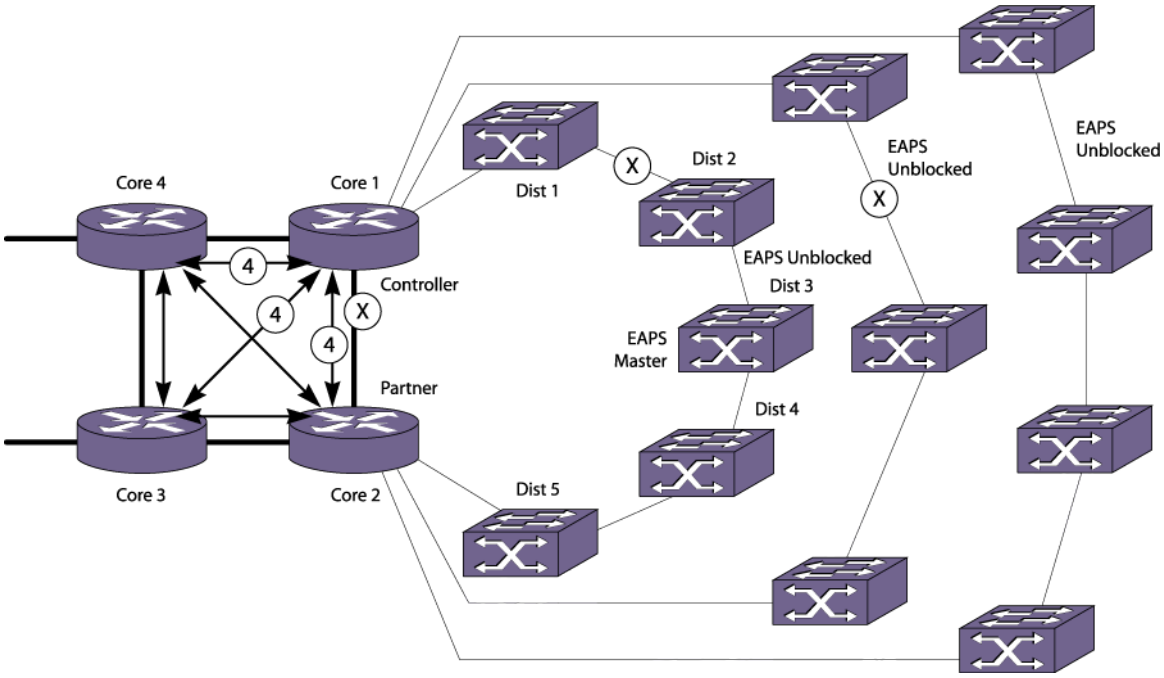


Figure 132: VPLS with Parallel Redundant EAPS Rings Configuration Example

PW Management for Network Failures

Table 77 shows how the controller node responds to multiple failures in a network with parallel EAPS rings.

Table 77: Controller Node Responses to EAPS Ring States

Ring State	Common Link Up	Common Link Down
Any parallel ring Up	PWs Active	PWs Inactive
All parallel rings Down	PWs Active	PWs Active

RSVP-TE Overview

RSVP is a protocol that defines procedures for signaling *QoS* requirements and reserving the necessary resources for a router to provide a requested service to all nodes along a data path. This release supports inter-area RSVP tunnels.

Note
Partial path computation for FRR over multiple *OSPF* areas is not supported.

RSVP is not a routing protocol. It works in conjunction with unicast and multicast routing protocols. An RSVP process consults a local routing database to obtain routing information. Routing protocols determine where packets get forwarded; RSVP is concerned with the QoS of those packets that are forwarded in accordance with the routing protocol.

Reservation requests for a flow follow the same path through the network as the data comprising the flow. RSVP reservations are unidirectional in nature, and the source initiates the reservation procedure by transmitting a path message containing a traffic specification (Tspec) object. The Tspec describes the source traffic characteristics in terms of peak data rate, average data rate, burst size, and minimum/maximum packet sizes.

RSVP-TE is a set of traffic engineering extensions to RSVP. RSVP-TE extensions enable RSVP use for traffic engineering in *MPLS* environments. The primary extensions add support for assigning MPLS labels and specifying explicit paths as a sequence of loose and strict routes. These extensions are supported by including label request and explicit route objects in the path message. A destination responds to a label request by including a label object in its reserve message. Labels are then subsequently assigned at each node the reserve message traverses. Thus, RSVP-TE operates in downstream-on-demand label advertisement mode with ordered LSP control.

The ExtremeXOS software implementation of RSVP-TE complies with RFC 3209 and includes support for:

- Configuration on a per *VLAN* interface.
- Operation as either edge or core MPLS router.
- Support for specifying explicitly routed paths.
- Support for both loose and strict route objects.
- Recording the route of an established path.
- Bandwidth reservation and policy per LSP.
- Signaling QoS along the RSVP path using the Tspec and Adspec objects.
- Fixed Filter (FF) and Shared Explicit (SE) reservation styles.
- Specifying RSVP-TE session attributes.
- Scaling enhancements using Refresh Overhead Reduction extensions.
- Improved link failure detection using the RSVP-TE Hello Message.
- Ability to reroute traffic over pre-configured backup LSPs.

RSVP Elements

Message Types

RSVP messages are passed between RSVP-capable routers to establish, remove, and confirm resource reservations along specified paths.

RSVP messages are sent as raw IP datagrams with protocol number 46. Each LSR along the path must process RSVP control messages so that it can maintain RSVP session state information. Therefore, most RSVP messages are transmitted with the IP Router Alert Option. Including the IP Router Alert provides a convenient mechanism allowing the IP routing hardware to intercept IP packets destined to a different IP address and deliver them to the RSVP control plane for processing. This is needed to set up and refresh RSVP-TE LSPs that follow an explicitly specified network path and thus may not use the normal routed next hop IP address. RSVP has two basic message types, path message and reserve message, as shown in the following figure.

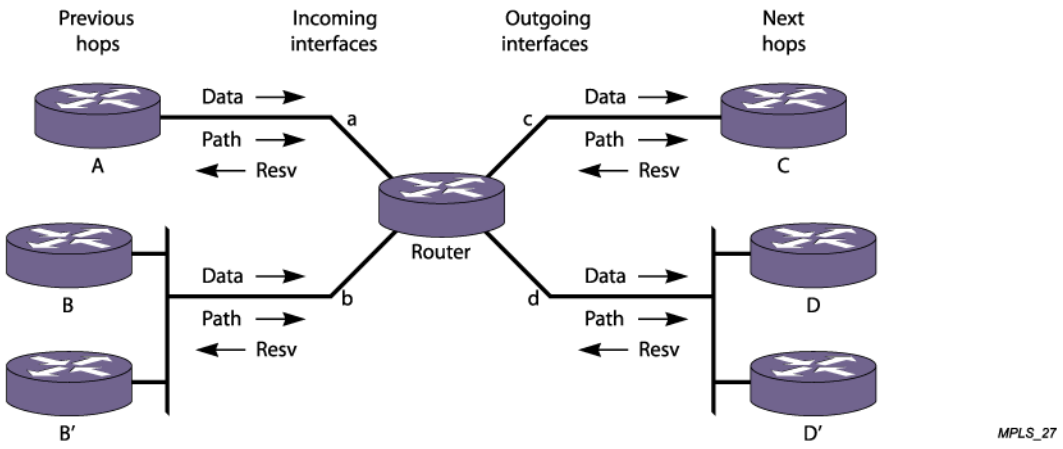


Figure 133: RSVP Messages

RSVP has the following message types:

- Path message
- Reserve message
- Path tear message
- Reserve tear message
- Path error message
- Reserve error message
- Reserve confirm message

Path Message: The RSVP path message is used to store state information about each node in the path. Each RSVP sender transmits path messages downstream along routed paths to set up and maintain RSVP sessions. Path messages follow the exact same path as the data flow, creating path states in each LSR along the path. The IP source address of the path message must be an address of the sender it describes and the IP destination address must be the endpoint address for the session. The path message is transmitted with the IP Router Alert option since each router along the path must process the path message. Each LSR is responsible for refreshing its path status by periodically transmitting a path message to the downstream LSR.

In addition to the previous hop address, the path message contains the sender Tspec and Adspec. The reservation message carries the flowspec.

Reserve Message: Each receiver host transmits an RSVP reservation request to its upstream neighbor. Reserve messages carry reservation requests hop-by-hop along the reverse path. The IP destination address of a reserve message is the unicast address of the previous-hop LSR, obtained from the session's path state. The IP source address is the address of the node that originated the message. The reserve message creates and maintains a reserve state in each node on the path. Each LSR is responsible for refreshing its reserve status by periodically transmitting a reserve message to the upstream LSR.

[6] The routed path may be the best routed path or an explicitly specified routed path using EROs.
[7] IP Router Alert option is described in RFC 2113.

Reserve messages are eventually delivered to the sender, so that the sender can configure appropriate traffic control parameters for the first hop node.

Path Tear Message: Path tear messages delete path state information reserved along the path. The message is initiated by the path sender or by any LSR in which a path state time-out occurs or an LSP is preempted (due to bandwidth reservations), and is sent downstream to the session's path endpoint. Path tear messages are transmitted with the IP Router Alert option and are routed exactly the same as path messages. The IP destination address must be the path endpoint and the source IP address must be the sender address obtained from the session's path state for the path that is being torn down.

When a path state is deleted as the result of the path tear message, the related reservation state must also be adjusted to maintain consistency in the node. The adjustment depends on the reservation style.

Reserve Tear Message: Reserve tear messages delete reservation state information. The message is initiated by the path endpoint or any node along the path in which a reservation state has timed out or an LSP is preempted (due to bandwidth reservations), and is sent upstream to the session's path sender. Reserve tear messages are routed exactly the same as reserve messages. The IP destination address of a reserve message is the unicast address of the previous-hop node, obtained from the session's reservation state. The IP source address is the address of the node that originated the message.

If no reservation state matches the reserve tear message, the message is discarded. The reserve tear message can delete any subset of the filter specification in FF-style or SE-style reservation state. Reservation styles are described in the following table.

Path Error Message: Path error messages are used to report processing errors for path messages. These messages are sent upstream to the sender that issued the path message. The message is routed hop-by-hop using the path state information maintained in each node. Path error messages are informational and do not modify the path state within any node.

Reserve Error Message: Reserve error messages are used to report processing errors for reserve messages. In addition, reserve error messages are used to report the spontaneous disruption of a reservation. Reserve error messages travel downstream to the endpoint of the session. The message is forwarded hop-by-hop using the reservation state information maintained in each node. Reserve error messages are informational and do not modify the reservation state within any node.

Reserve Confirm Message: Reserve confirm messages are optionally transmitted to acknowledge a reservation request. These messages are transmitted from the sender to the endpoint. The destination IP address is the IP address of the endpoint and the source IP address is the address of the sender. Since none of the intermediate path nodes need to process a reserve confirm message, the message is transmitted without the IP Router Alert option.

Reservation Styles

A reservation style is a set of options that is included in the reservation request.

One reservation style concerns how reservations requested by different senders within the same session are handled. This type of reservation style is handled in one of two ways: either create a distinct reservation for each sender in the session, or use a single reservation that is shared among all packets of the selected senders.

Another reservation style concerns how senders are selected. Again, there are two choices: an explicit list of all selected senders, or a wildcard that implies all senders in the session.

Table 78 describes the relationship between reservation attributes and styles.

Table 78: Reservation Attributes and Styles

Sender Selection	Distinct Reservation Style	Shared Reservation Style
Explicit	Fixed filter (FF)	Shared explicit (SE)
Wildcard	Not defined	Wildcard filter (WF)

Fixed Filter: The fixed filter (FF) reservation style uses a distinct reservation and an explicit sender selection. This means that each resource reservation is for a specific sender. The session resources are not shared with other senders' packets. Because each reservation is identified with a single sender, a unique label is assigned by the endpoint to each sender (i.e., point-to-point LSP reservation).

Shared Explicit: The shared explicit (SE) reservation style uses a shared reservation and an explicit sender selection. This means that a single resource reservation is created that is shared by multiple senders. The endpoint may specify which senders are to be included for the reservation. Because different senders are explicitly listed in the RESV message, different labels may be assigned to each sender. Thus, multiple shared-resource LSPs to the same endpoint can be created (i.e., multipoint-to-point LSP reservation). The Extreme *MPLS* implementation requests SE reservation style when signaling RSVP-TE LSPs.

Wildcard: The wildcard (WF) reservation style uses the shared reservation and wildcard sender options. A wildcard reservation creates a single reservation that is shared by data flows from all upstream senders.

The Extreme MPLS implementation does not support WF reservation style.

RSVP Traffic Engineering

MPLS Traffic Engineering (TE) extends RSVP to support several unique capabilities.

By coupling RSVP and MPLS, LSPs can be signaled along explicit paths with specific resource reservations. Additional RSVP objects have been defined to provide TE extensions. These objects include the Label Request, Label, Explicit Route, Record Route, and Session Attribute. Extreme's RSVP-TE implementation supports all of these TE objects.

RSVP Tunneling

An RSVP tunnel sends traffic from an ingress node through an LSP. The traffic that flows through the LSP is opaque (or tunneled) to the intermediate nodes along the

path. Traffic flowing through the tunnel to an intermediate node along the path is identified by the previous hop and is forwarded, based on the label value(s), to the downstream node.

RSVP-TE can:

- Establish tunnels with or without *QoS* requirements.
- Dynamically reroute an established tunnel.
- Observe the actual route traversed by a tunnel.
- Identify and diagnose tunnels.
- Use administrative policy control to preempt an established tunnel.
- Perform downstream-on-demand label allocation, distribution, and binding.

Some LSRs require their neighboring LSRs to include their Router ID in the Extended Tunnel ID field when sending RSVP-TE messages. The Extended Tunnel ID is a globally unique identifier present in the RSVP common header Session object (see RFC 3209). To provide maximum compatibility with other vendors' implementations, the ExtremeXOS *MPLS* implementation accepts RSVP-TE messages regardless of the Extended Tunnel ID value and always inserts the local Router ID into the Extended Tunnel ID field prior to transmission of an RSVP-TE message.

RSVP Objects

This section describes the RSVP objects that are used to establish RSVP-TE LSPs:

- Label
- Label request
- Explicit
- Record route
- Session attribute

Label: The label object is carried in the reserve message and is used to communicate a next hop label for the requested tunnel endpoint IP address upstream towards the sender.

Label Request: To create an RSVP-TE LSP, the sender on the *MPLS* path creates an RSVP path message and inserts the label request object into the path message.

A label request object specifies that a label binding for the tunneled path is requested. It also provides information about the network layer protocol that is carried by the tunnel. The network layer protocol sent through a tunnel is not assumed to be IP and cannot be deduced from the Layer 2 protocol header, which simply identifies the higher layer protocol as MPLS. Therefore, the Layer 3 Protocol ID (PID) value must be set

in the Label Request Object, so that the egress node can properly handle the tunneled data.

Note
The ExtremeXOS RSVP-TE implementation supports only Label Request objects with no Label Range. Label Ranges are used to signal ATM VPI/VCI or Frame Relay DLCI information for the LSP. These types of Label Requests are not supported. In the ExtremeXOS RSVP-TE implementation, the L3 PID value, which identifies the Layer 3 protocol of the encapsulated traffic, is always set to 0x0800 (IP).

Explicit Route: The explicit route object specifies the route of the traffic as a sequence of nodes. Nodes may be loosely or strictly specified.

The explicit route object is used by the MPLS sender if the sender knows about a route that:

- Has a high likelihood of meeting the *QoS* requirements of the tunnel.
- Uses the network resources efficiently.
- Satisfies policy criteria.

If any of the above criteria are met, the sender can decide to use the explicit route for some or all of its sessions. To do this, the sender node adds an explicit route object to the path message.

After the session has been established, the sender node can dynamically reroute the session (if, for example, if discovers a better route) by changing the explicit route object.

Record Route: The record route object is used by the sender to receive information about the actual route traversed by the RSVP-TE LSP. It is also used by the sender to request notification if there are changes to the routing path. Intermediate or transit nodes can optionally use the RRO to provide loop detection.

To use the object, the sender adds the record route object to the path message.

Session Attribute: The session attribute object can also be added to the path message. It is used for identifying and diagnosing the session. The session attribute includes the following information:

- Setup and hold priorities
- Resource affinities
- Local protection

ERO Exclude Option

In order to allow more flexibility when traffic engineering redundant paths for RSVP-TE LSPs, this feature adds the “exclude” option to the “configure mpls rsvp-te path <path> add ero” command. An “include” option is also available, but is optional. If neither option is used, “include” will be used for backward compatibility.

A hop that is excluded is avoided when CSPF does path calculations. By adding an “exclude” hop to a path, LSPs can be set up to avoid certain links. An example of

this is to configure a secondary RSVP-TE LSP to avoid the hops that a primary LSP has been configured to traverse. This can allow the secondary LSP more freedom to route through other parts of the network. Often, without the “exclude” option, the secondary LSP must be configured more tightly than desired to ensure that its path never overlaps with the primary LSP’s path.

Establishing RSVP-TE LSPs

Establishing LSPs requires every LSR along the path to support RSVP and the TE extensions defined in RFC 3209.

The LSP endpoints attempt to detect non-RSVP capable LSRs by comparing the time-to-live (TTL) value maintained in the RSVP common header with that of the IP TTL. If these values are different, it is assumed that a non-RSVP capable LSR exists along the path. By including the Label Request object in the path message, RSVP capable routers that do not support the TE extensions can be detected. RSVP routers that do not support TE extensions reply with the Unknown object class error.

RSVP-TE LSPs are referred to as named LSPs. These LSPs have configurable names that are used to identify the LSP within the CLI. The command `create mpls rsvp-te lsp` *`lsp_name`* `destination` *`ipaddress`* allocates the internal resources for the LSP. The newly created LSP is not signaled until the LSP has been configured. The LSP can be configured to take a specific path through the network or the administrator can let the switch choose the best path by specifying the path any. Up to three paths may be configured for an LSP to provide redundancy. The command `configure mpls rsvp-te lsp` *`lsp_name`* `add path` configures an LSP. Optionally, RSVP-TE profiles may be applied to an LSP to change its properties. An RSVP-TE profile is a specific CLI container used to hold configuration parameters associated with timers, bandwidth reservation, limits, and other miscellaneous properties.

Once the RSVP-TE LSP is configured, the LSP is immediately signaled. If signaled successfully, the LSP becomes active. The commands `disable mpls rsvp-te lsp` *`lsp_name`* and `enable mpls rsvp-te lsp` *`lsp_name`* are used to tear down and re-signal the LSP. Disabling the LSP causes the LER to send a path tear message to the destination, forcing the LSP down and all resources along the path to be freed. Enabling the LSP instructs the LER to send a path message to the destination re-establishing the LSP. The configuration of the LSP is not modified by the enable or disable LSP commands.

RSVP-TE Implementation

Explicit Route Path LSPs

An explicit route is a specified path through a routed network topology.

The path can be strictly or loosely specified. If strictly specified, each node or group of nodes along the path must be configured. Thus, no deviation from the specified path is allowed.

Loosely specified paths allow for local flexibility in fulfilling the requested path to the destination. This feature allows for significant leeway by the LSR in choosing the next

hop when incomplete information about the details of the path is generated by the LER. Each node along the path may use other metrics to pick the next hop along the path, such as bandwidth available, class of service, or link cost. The command configure mpls rsvp-te path *path_name* add ero is used to add an Explicit Route Object to a path container.

An explicit routed path is encoded using the explicit route object (ERO) and is transmitted in the path message. The ERO consists of a list of subobjects, each of which describes an abstract node. By definition, an abstract node can be an IP prefix or an autonomous system (AS) number. The ExtremeXOS RSVP-TE implementation supports only IPv4 abstract nodes. The ExtremeXOS RSVP-TE implementation supports both strict and loose IPv4 abstract nodes. Received path messages with EROs that contain any other subobject type result in the transmittal of an Unknown object class error message. All LSRs along the specified path must support the inclusion of the ERO in the path message for an explicitly routed path to be successfully set up.

An LSR receiving a path message containing an ERO must determine the next hop for this path.

The steps for selection of the next hop are as follows:

1. The receiving LSR evaluates the first subobject. If the subobject type is not supported or there is no subobject, a Bad ERO error is returned. The abstract node is evaluated to ensure that this LSR was the valid next hop for the path message. If the subobject is a strict abstract node, the abstract node definition must match the local interface address. If it does, then this LSR is considered to be a member of the abstract node. Additionally, if the /32 address matches a local interface address, the path message must have been received on the direct interface corresponding to the /32 address. If the abstract node is an IP prefix, the subnet configured for the interface from which the path message was received must match the abstract node definition. In the event that this LSR is not part of the strict abstract node definition, a Bad initial subobject error is returned. If the subobject is a loose abstract node, the LSR determines if the abstract node definition corresponds to this LSR. If it doesn't, the path message is transmitted along the best-routed or constrained optimized path to the endpoint and the ERO is not modified. If it is, then processing of the ERO continues.
2. If there is no second subobject, the ERO is removed from the path message. If this LSR is not the end of the path, the next hop is determined by the constrained optimized path (through Constrained Shortest Path First—CSPF) to the path message endpoint.
3. If there is a second subobject, a check is made to determine if this LSR is a member of the abstract node. If it is, the first subobject is deleted and the second subobject becomes the first subobject. This process is repeated until either there is only one subobject or this LSR is not a member of the abstract node as defined by the second subobject. Processing of the ERO is then repeated with step 2. By repeating steps 2 and 3, any redundant subobjects that are part of this LSRs abstract node can be removed from the ERO. If this operation were not performed, the next hop LSR might reject the path message.
4. The LSR uses its CSPF to determine the next hop to the second subobject. If the first object is a /32 address, the first subobject is removed, since it would not be part of

the next hop's abstract node. The path message is then sent along the explicit path to the path message endpoint. No determination is made to verify that the abstract node defined in the subobject is topologically adjacent to this LSR. The next hop should verify this as part of its processing as defined in step 1.

If CSPF determines that a specific path needs to be taken through the network, additional EROs are inserted into the path message.

Route Recording

The route a path takes can be recorded.

Recording the path allows the ingress LER to know, on a hop-by-hop basis, which LSRs the path traverses. Knowing the actual path of an LSP can be especially useful for diagnosing various network issues.

Network path recording is configurable per LSP. This feature is configured by enabling route recording for a specific RSVP-TE profile using the command `configure mpls rsvp-te lsp profile` *`lsp_profile_name`* `record enabled` and associating the profile to an LSP. The ExtremeXOS software sets the label recording desired flag in the path message if route recording has been enabled for the LSP.

If route recording is enabled, the record route object (RRO) is inserted into the path message using a single RRO subobject, representing the ingress LER. When a path message that contains an RRO is received by an Extreme LSR, an RRO IPv4 subobject representing the /32 address of the outgoing interface of the path message is pushed onto the top of the first RRO. The updated RRO is returned in the reserve message.

The label recording flag is supported by the ExtremeXOS software and is set automatically when route recording is enabled. The route-only option can be used when enabling route recording in the profile to prevent the label recording flag from being set. If an Extreme LSR receives a path message with the label recording flag set in the RRO, the LSR encodes the LSP label into a label subobject and pushes it onto the RRO.

If a path message is received that contains an RRO, the Extreme LSR uses the RRO to perform loop detection. The RRO is scanned to verify that the path message has not already traversed this LSR. If the RRO contains an IPv4 subobject that represents a local LSR interface, the path message is dropped and a Routing Problem error message is sent to the originating LER with an error value of Loop detected.

LSP Session Attributes

Session attributes are signaled for configured RSVP-TE LSPs using the session attribute object without resource affinities (that is, LSP_TUNNEL Type).

The ExtremeXOS software uses the setup and hold priority values to preempt established LSPs in order to satisfy bandwidth requests. Lower hold priority LSPs are preempted in order to satisfy the bandwidth request in a path message with a higher setup priority. LSP attributes are configured by setting the priorities for

[8] RRO is organized as a LIFO stack.

a specific RSVP-TE profile using the command `configure mpls rsvp-te lsp profile lsp_profile_name setup-priority priority hold-priority priority` and associating the profile to the configured LSP.

Bandwidth Reservation

As mentioned previously, RSVP reservations are unidirectional in nature.

The source initiates the reservation procedure by transmitting a path message containing a sender Tspec object. The Tspec describes the source traffic characteristics in terms of peak data rate, average data rate, burst size, and minimum/maximum packet sizes. The path message can also contain an optional AdSpec object that is updated by network elements along the path to indicate information such as the availability of particular *QoS* services, the maximum bandwidth available along the path, the minimum path latency, and the path maximum transmission unit (MTU).

The ExtremeXOS software supports LSR bandwidth reservation requests per LSP. Only the Int-Serv Controlled-Load service request is supported. Bandwidth is always reserved on the physical ports that the LSP traverses. Depending on the platform, the bandwidth reservation may also be policed. The network administrator can verify that the requested bandwidth was actually reserved. In those cases when the bandwidth reserved is less than the requested bandwidth, the LSP can be manually torn down, re-signaled using a different path, or accepted. The LSR automatically attempts to find a path that best satisfies the bandwidth request. Constrained path selections are supported using *OSPF*-TE. Best effort LSPs are provisioned by specifying a reserved bandwidth as best-effort. The reserved LSP bandwidth is configured by setting the bps rate for a specific RSVP-TE profile, using the `configure mpls rsvp-te lsp profile lsp_profile_name bandwidth` command and associating the profile to an LSP.

Accounting of bandwidth reserved through an Extreme LSR RSVP-TE enabled *VLAN* is supported. The maximum available bandwidth per physical port or trunk group is enforced. Thus, the available bandwidth specified in the Adspec object is not modified as the path message is forwarded to the LSP endpoint. As reserve messages are processed, the reserved bandwidth specified in the Flowspec is added to the total reserved bandwidth allocated for the physical ports.

Because LSP bandwidth is dynamically allocated, a configuration command is provided to reserve port bandwidth for use by *MPLS*. The command **`configure mpls rsvp-te bandwidth committed-rate`** pre-reserves bandwidth from the specified MPLS enabled VLAN for RSVP-TE traffic only. This pre-allocation of bandwidth is useful since other applications may compete with MPLS for available bandwidth. By pre-reserving a portion of the MPLS interface's bandwidth capacity, MPLS is guaranteed to have that amount of the MPLS interface's bandwidth to meet RSVP-TE LSP reservation requests.

CIR bandwidth for the receive direction is not tracked by TE IGPs, such as OSPF-TE, and configuring it is not required. Configuring CIR bandwidth for the receive direction does not prevent an LSP from going operational due to lack of receive bandwidth; however, it can be useful for tracking and informational purposes. An Info level log (MPLS.RSVPTE.IfRxBwdthExcd) is generated if the setup of a TE LSP requires receive bandwidth greater than that which is currently available for the receive direction on a particular interface. This generally happens only when TE LSPs with different previous

hops ingress the switch on the same interface (for example, from a multi-access link) and egress the switch on different interfaces.

Bandwidth Management for RSVP-TE LSPs

If an RSVP-TE LSP is signaled through a switch with bandwidth parameters, the LSP bandwidth request is granted or rejected based on the availability of bandwidth resources on the physical ports that the LSP traverses.

Data traffic through these switches is not policed and there are no guarantees that the packets using the LSP are not dropped.

Note
Per LSP rate limiting is not supported in this release.

The available bandwidth for each *OSPF* interface is continually updated within the OSPF area. As RSVP-TE LSPs are established and torn down, the reserved bandwidth associated with these LSPs is used to update the total bandwidth available through each OSPF interface. RSVP-TE and CSPF can use the bandwidth information to determine the appropriate path that each LSP should take through the network based on the LSP's profile parameters. LSP parameters that can affect the CSPF TE path calculation include the LSP setup priority and bandwidth configuration.

Available bandwidth is calculated for eight *CoS (Class of Service)* levels. Each CoS uniquely maps to an LSP hold priority. Thus, when an LSP is set up through the switch, the reserved bandwidth consumed is associated with a CoS based on the signaled LSP hold priority. The available bandwidth is recalculated and is advertised to its OSPF neighbors. Advertised bandwidth is calculated using graduated bandwidth reporting methodology. Using this scheme, higher CoS levels advertise available bandwidth that includes allocated bandwidth for lower CoS levels. The reasoning for doing this is that higher priority LSPs can preempt lower priority LSP. Thus, even though the bandwidth has been allocated to a lower priority LSP, it is still available for use by higher priority LSPs.

In the following example, an interface is configured to reserve 250 Mbps for *MPLS* traffic.

The following LSPs are established through this interface. Remember, hold priority value of 0 is the highest priority and 7 is the lowest.

- LSP A, hold priority = 7, reserved = 50 Mbps
- LSP B, hold priority = 5, reserved = 100 Mbps
- LSP C, hold priority = 2, reserved = 25 Mbps
- LSP D, hold priority = 1, reserved = 25 Mbps

OSPF advertises the following available bandwidth for each CoS.

CoS 0 is the highest and CoS 7 is the lowest:

- CoS 0 (hold = 0): 250 Mbps (No LSPs; all bandwidth available)
- CoS 1 (hold = 1): 225 Mbps (LSP D)
- CoS 2 (hold = 2): 200 Mbps (LSP C & D)

- CoS 3 (hold = 3): 200 Mbps (LSP C & D)
- CoS 4 (hold = 4): 200 Mbps (LSP C & D)
- CoS 5 (hold = 5): 100 Mbps (LSP B, C & D)
- CoS 6 (hold = 6): 100 Mbps (LSP B, C & D)
- CoS 7 (hold = 7): 50 Mbps (LSP A, B, C & D)

CSPF calculations only use the available bandwidth for the desired CoS, as specified by the LSP hold priority. Thus in this example, if LSP E, with a configured setup priority of 6, requires 150 Mbps, CSPF calculates a path to the destination that does not go through the above interface, since only 100 Mbps worth of bandwidth is available.

Redundant LSPs

There are three methods for provisioning redundant RSVP-TE LSPs at the ingress LER, also referred to as head-end LSP protection:

- Configured secondary (or backup) LSPs
- Fast reroute (detour) LSPs
- Multipath LSPs

Secondary RSVP-TE LSPs can be configured to provide backup LSPs in the event that the primary LSP fails. You can create up to two secondary LSPs for each primary LSP. The secondary LSPs are fully provisioned, pre-established RSVP-TE LSPs that are maintained as inactive until needed. If the primary LSP is torn down, the associated LSP next hop is removed from the route table, and a new LSP next hop representing one of the secondary LSPs is installed as the preferred LSP. If there are multiple secondary LSPs available, the secondary LSP is randomly selected. If the primary LSP is re-established, the primary LSP next hop information is re-installed and the secondary LSP returns to inactive state.

If both the primary and secondary paths for an LSP fail, and there are no other RSVP-TE LSPs active to the destination, an LDP LSP can be used if available.

Operation with L2 VPNs is similar. If a primary path fails, and a secondary LSP is available, VPLS uses the secondary LSP. When the primary LSP is re-established, VPLS again uses the primary LSP.

Specifying redundant LSPs is accomplished by assigning secondary paths to an LSP. The `configure mpls rsvp-te lsp` *`lsp_name`* `add path` *`path_name`* `secondary` command can configure the specified path as a backup LSP. A path different from the primary path must be specified. It is recommended that defined paths be configured using EROs to specify different paths through the network. Relying on the routing topology, by configuring the path to any, can create two LSPs that take the same path. It is important to understand that the configured LSP signals multiple LSPs, up to three (one primary and two secondary), but only one LSP can be used to forward traffic at any one time.

Fast Reroute LSPs are based on the on IETF RFC 4090, Fast Reroute Extensions to RSVP-TE for LSP Tunnels, which defines RSVP-TE extensions to establish backup LSP tunnels for local repair of LSP tunnels. To respond to failures, these mechanisms enable

the re-direction of traffic onto backup LSP tunnels in tens of milliseconds, and this meets the needs of real-time applications such as voice over IP (VoIP). This timing requirement is satisfied by computing and signaling backup LSP tunnels in advance of a failure and by re-directing traffic as close to the failure point as possible. In this way the time for redirection includes no path computation and no signaling delays, which include delays to propagate failure notification between label-switched routers (LSRs). Speed of repair is the primary advantage of using fast-reroute backup methods.

There are two backup methods; the detour LSP method (which is also called the one-to-one backup method) and the facility backup method (which is also called the by-pass tunnel method). The software supports only the detour LSP method.

Based on the RFC-4090 there are two different methods to uniquely identify a backup path:

1. Path-specific method
2. Sender template-specific method

The software supports only the path-specific method, which uses a new object, the DETOUR object, to distinguish between PATH messages for a backup path and the protected LSP.

Figure 134 illustrates the terminology used to describe fast-reroute configuration and operation.

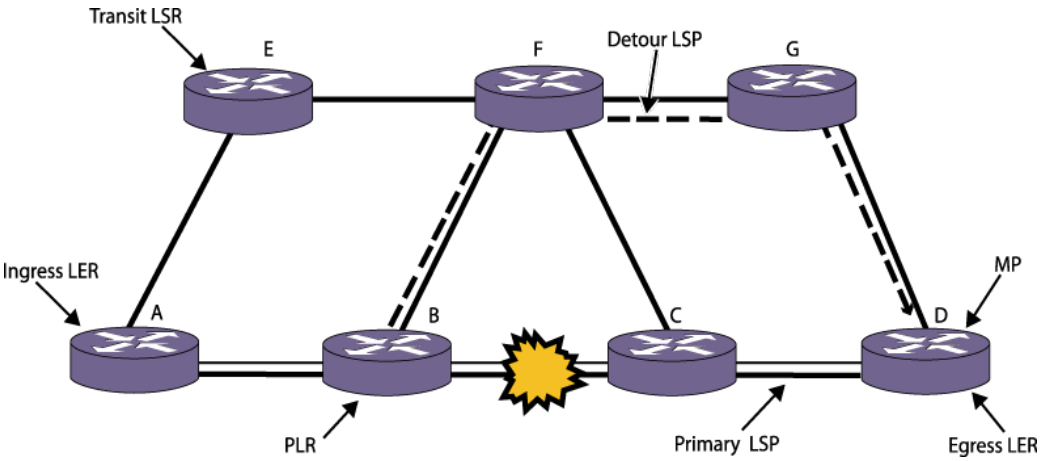


Figure 134: Fast-Reroute Terminology

The primary LSP in Figure 134 is established between *MPLS* routers A and D. Router A is the ingress LER, and Router D is the egress LER. When used with fast-reroute protection, the primary LSP is also called the protected LSP, as it is protected by the detour LSP created by the fast-reroute feature. The detour LSP provides a route around a protected component. The link between Router B and Router C has failed. The detour LSP, which is indicated by the dashed line, provides a path around the failure.

Routers B and C are transit LSRs for the primary LSP. With respect to a specific LSP, any router that is not the ingress or egress LER is a transit LSR. Routers F and G are transit LSRs for the detour LSP.

The origin of the detour LSP is called the Point of Local Repair (PLR), and the termination of the detour LSP is called the Merge Point. A protected LSP is an explicitly-routed LSP that is provided with protection. A detour LSP is also an explicitly-routed LSP. If you configure a series of one or more hops (EROs), then based on the currently set DYNAMIC_FULL option in the Constrained-based Shorted Path First (CSPF) routing component, the CSPF will calculate and try to fill in the gaps to build a complete list of EROs.

You can configure up to two secondary LSPs for each standard TE (non-FRR) LSP or for each protected FRR LSP. If a standard TE LSP fails, then one of the secondary LSPs becomes active. If that secondary LSP fails, the other secondary LSP becomes active. If a protected FRR LSP fails, its detour LSP becomes active. If the detour LSP fails, then one of the secondary LSPs becomes active, and if that secondary LSP fails, the other secondary LSP becomes active. If all configured backup and secondary paths for an LSP fail, a different active RSVP-TE LSP to the destination can be used. Otherwise, an LDP LSP can be used if available.

The primary advantage of detour LSPs is the repair speed. The cost of detour LSPs is resources. Each backup LSP reserves resources that cannot be used by other LSPs. Another cost is that currently there is no automatic way to redirect traffic from a detour LSP back to a primary LSP when the protected LSP recovers. Redirecting traffic from the detour LSP to the primary LSP requires a series of CLI commands.

Fast reroute protection is configured primarily on the ingress LER, however, it must be enabled on all transit LSRs and the egress LER also. After configuration is complete and fast-reroute protection is enabled on the primary LSP, the primary and detour LSPs are signalled. Provided that the resources are available, detour LSPs are set up at each transit LSP along the primary LSP.

Multiple RSVP-TE LSPs can exist or be configured to the same destination. The paths do not need to be equal cost; all that is required is that all the LSPs to the same destination must have IP transport enabled. In this scenario, LSP next hop information is communicated to the route table for up to eight different named RSVP-TE LSPs. Locally originated traffic is distributed across each LSP based on standard IP address hash algorithms. If one of the LSPs fails, the traffic is redistributed across the remaining active named LSPs. Unlike the backup LSP mechanism, all of the redundant multipath LSPs are unique named LSPs and in general have primary configured paths.

Improving LSP Scaling

RSVP maintains path and reserve state by periodically sending refresh messages.

Refresh messages allow each LSR along the path to properly maintain reservation state information and to recover from network failures. Because refresh messages are periodically sent for each path reservation, scaling the number of RSVP-TE LSPs is an issue. Additionally, network requirements for faster failure detection and improved LSP recovery times further exacerbate the scaling issue.

Several techniques are described in RFC 2961 RSVP Refresh Overhead Reduction to improve the scalability of RSVP. These techniques include the bundle message, message ID extension, and summary refresh extension. Support for these extensions is signaled between RSVP peers via the refresh-reduction-capable bit in the flags field

of the common RSVP header. Additionally, the hello extension, described in RFC 3209, provides a fourth scaling mechanism for RSVP. The hello extension is designed so that either peer can use the mechanism regardless of how the other peer is configured. Therefore, support for the hello extension is not signaled between RSVP peers. The ExtremeXOS software supports and is compliant with the RSVP-TE scaling features described in RFC 2961.

These features include the following:

- Bundle message
- Summary refresh extension
- Message ID extension
- Hello extension

Bundle Message: RSVP bundle messages aggregate multiple RSVP messages within a single PDU. The messages are addressed directly to peer LSRs. Therefore, bundle messages are not sent with the IP Router Alert option. Bundling multiple RSVP messages into a single PDU reduces the per packet overhead associated with local delivery and local origination. Each bundle message must contain at least one RSVP message. Transmission of RSVP messages may be delayed up to the number of seconds configured for bundle time. The size of the bundle message is limited to the RSVP-TE interface MTU size. Bundle messaging is enabled using the enable mpls rsvp-te bundle-message command.

Summary Refresh Extension: A summary refresh message is used to refresh RSVP states along an LSP without having to explicitly send path and reserve refresh messages. This can substantially reduce the RSVP control bandwidth overhead. Summary refresh messages contain a list of message_ID objects. Each message_ID object identifies a path and reserve state to be refreshed. When summary refresh support is enabled, path and reserve refresh messages are suppressed. If the message identifier value indicates that the RSVP state has changed, the receiving LSR notifies the sender by transmitting a message_ID_NACK message. The summary refresh rate is enabled using the enable mpls rsvp-te summary-refresh command.

Message ID Extension: The message ID extension provides reliable delivery of RSVP messages. It also provides a simple mechanism for identifying refresh messages, which can greatly reduce refresh message processing on the receiving LSR. The message ID extension defines three new objects: message_ID, message_ID_ACK, and message_ID_NACK. The message_ID object contains a unique message identifier based on the sender's IP address. Only one message_ID object is inserted into an RSVP message. The receiving LSR can use the message_ID object to quickly refresh path and reserve states. If the message identifier value in the message_ID object is greater than the locally saved message identifier value, then the RSVP message represents a new or modified state. The receiving LSR must acknowledge an RSVP message using the message_ID_ACK object if the sender set the ACK_desired flag in the message_ID object, otherwise the message_ID acknowledgement is optional. The message_ID_ACK object may be included in any unrelated RSVP message or in an RSVP ACK message. Message ID extension is required for both bundle message and summary refresh, so this capability is automatically enabled if either of the other capabilities is enabled.

Hello Extension: The RSVP hello message provides a quick and simple mechanism for detecting the loss of a peer RSVP-TE LSR. The hello protocol is implemented using the RSVP soft-state model. RSVP hello messages may be enabled independently of each LSR peer. The hello protocol consists of two new objects: hello_request and hello_ACK. If configured, an LSR sends a hello_request every hello interval. If a hello_ACK is not received within a specified amount of time, the sending LSR assumes that its peer LSR is no longer active. Once a peer LSR is deemed inactive, all reservation states associated with LSPs established to or through the peer LSR must be freed and the LSPs torn down. The hello interval is configurable using the command configure mpls rsvp-te timers session hello-time.

You can improve LSP scaling by configuring the following RSVP-TE parameters:

- Refresh time
- Summary refresh time
- Bundle time

Refresh Time: The refresh time specifies the interval for sending refresh path messages. RSVP refresh messages provide soft state link-level keep-alive information for previously established paths and enable the switch to detect when an LSP is no longer active. RSVP sessions are torn down if an RSVP refresh message is not received from a neighbor within [(keep-multiplier + 0.5) * 1.5 * refresh-time] seconds. The valid refresh time may be set to any value between 1 and 36000 seconds. The default setting is 30 seconds. Configuring a longer refresh time reduces both switch and network overhead.

Summary Refresh Time: The summary refresh time, specified in tenths of a second, indicates the time interval for sending summary refresh RSVP messages. The summary refresh time must be less than the configured refresh time. The default summary refresh time is zero, indicating that no summary refresh RSVP messages are sent. The summary refresh time value may be set to any value between zero to 100 (or 10 seconds). If configured, the bundled and summary refresh RSVP messages are only sent to RSVP-TE peers supporting RSVP refresh reduction.

Bundle Time: The bundle time, specified in tenths of a second, indicates the maximum amount of time a transmit buffer is held so that multiple RSVP messages can be bundled into a single PDU. The default bundle time is zero, indicating that RSVP message bundling is not enabled. The bundle time value can be set to any value between zero and 30 (or 3 seconds).

Supporting Quality of Service Features

QoS LSP support is an important attribute of *MPLS*.

MPLS supports the Differentiated Services (DiffServ) model of QoS. The DiffServ QoS model is supported by mapping different traffic classes to different LSPs, or by using the EXP bits in the MPLS shim header to identify traffic classes with particular forwarding requirements.

Propagation of IP TTL

There are two modes of operation for routed IP packets: pipe TTL mode and uniform TTL mode.

Currently, switches that run Extreme OS support only the pipe TTL mode. In pipe TTL mode, the LSP is viewed as a point-to-point link between the ingress LSR and the egress LSR; intermediate LSRs in the *MPLS* network are not viewed as router hops from an IP TTL perspective. Thus, the IP TTL is decremented once by the ingress LSR, and once by the egress LSR.

In this mode, the MPLS TTL is independent of the IP TTL. The MPLS TTL is set to 255 on all packets originated by the switch and on all packets that enter a pseudowire.

Configuring MPLS

MPLS has the following configuration constraints:

- **IP flow redirection**—IP flow redirection commands and MPLS are mutually exclusive functions. Both functions cannot be enabled simultaneously.
- ***IGMP* snooping**—*OSPF* and LDP session establishment require the switch to receive and process IP multicast frames. Therefore, IGMP snooping must be enabled to support MPLS.
- **VPLS**—VPLS requires that IGMP snooping be disabled on customer-facing VLANs.

Configuration Overview

Do the following to configure MPLS:

1. If you want to use in a different VR (the default is VR-Default), move MPLS to the appropriate VR as described in Moving MPLS From VR to VR.
2. Create and configure the VLANs that will use MPLS.
3. Configure an MPLS LSR ID for each switch that serves as an LSR in the network. (See Configuring the MPLS LSR ID).
4. Add MPLS support to the appropriate VLANs. (See Adding MPLS Support to VLANs).
5. Enable MPLS on each switch that serves as an LSR. (See Enabling and Disabling MPLS on an LSR).
6. Enable MPLS on the appropriate VLANs. (See Enabling and Disabling MPLS on a VLAN).
7. Enable LDP on each switch that serves as an LSR. (See Enabling LDP on the Switch).
8. Enable LDP on the appropriate VLANs. (See Enabling and Disabling LDP on a VLAN).
9. Configure an IGP, such as or IS-IS, for the appropriate switches and VLANs.

Moving MPLS From VR to VR

By default, *MPLS* is enabled on the *VR-Default* . You can operate MPLS in VR-Default or in any user VR, but MPLS only operates in one VR. If you want to use MPLS in a user VR, you must delete it from VR-Default first.

- To delete MPLS from a VR, do the following:
  a. Disable MPLS on the LSR as described in Enabling and Disabling MPLS on an LSR on page 1108.
  b. Disable MPLS on all VLANs and reset the MPLS configuration as described in Resetting MPLS Configuration Parameter Values on page 1116.
  c. Remove MPLS from the VR by entering the following command:

  ```
  configure vr vr-name delete protocol protocol-name
  ```

  Note
  When you enter the command to delete MPLS from a VR, the software displays a prompt to remind you that the MPLS configuration is lost when MPLS is deleted.

- To add MPLS to VR, do the following:
  a. Use the following command:

  ```
  configure vr name add protocol mpls
  ```

  b. Add any other protocols that you want to use with MPLS.

After you add MPLS to a VR, you must configure and enable MPLS. MPLS commands operate only in the VR to which MPLS is added. An error message appears if you enter an MPLS command in a VR that does not support MPLS.

Configuring the MPLS LSR ID

The *MPLS* LSR ID must be configured before MPLS can be used. The address chosen must be a routable IP address on the switch. It is suggested that this be set to the same IP address as a routing protocol ID (for example, the *OSPF* Router ID).

- To configure the MPLS LSR ID, use the following command:

  ```
  configure mpls lsr-id ipaddress
  ```

  Note
  The MPLS LSR ID must be configured before MPLS can be enabled. The LSR ID should be configured on a loopback *VLAN*.

Adding MPLS Support to VLANs

To use *MPLS* on a *VLAN*, MPLS must first be configured on that VLAN.

- To configure a specific VLAN or all VLANs, use the following command:

  ```
  configure mpls add {vlan} vlan_name
  ```

MPLS must be configured on a VLAN before it can be used to transmit or receive MPLS-encapsulated frames. By default, MPLS is not configured on a newly created VLAN.

If you have enabled MPLS on an *OSPF* interface that is used to reach a particular destination, make sure that you enable MPLS on all additional OSPF interfaces that can reach that same destination (for example, enable MPLS on all VLANs that are connected to the backbone network).

Enabling and Disabling MPLS on an LSR

- To enable *MPLS* on an LSR, use the following command:

  ```
  enable mpls
  ```

  Note
  For SummitStacks, see the Switch Engine v33.6.1 Command References description for the `enable mpls` command for special requirements for SummitStacks.

  By default, MPLS is disabled on the switch. This command starts the MPLS process, allowing MPLS protocols to run and MPLS packets to be forwarded.
- To disable MPLS on an LSR, use the following command:

  ```
  disable mpls
  ```

  When you disable MPLS, LDP and RSVP-TE are effectively disabled. The MPLS configuration remains intact.

Enabling and Disabling MPLS on a VLAN

After *MPLS* is enabled globally, MPLS must be enabled on the MPLS configured *VLAN*s. Configuring a VLAN for MPLS does not enable MPLS on the VLAN. The VLAN must be specifically MPLS enabled in order to send and receive MPLS packets over an interface.

- To enable MPLS on specific VLANs, use the following command:

  ```
  enable mpls [{vlan}vlan_name|vlan all]
  ```
- To disable MPLS on specific VLANs, use the following command:

  ```
  disable mpls [{vlan} vlan_name | vlan all]
  ```

  MPLS must be enabled on all VLANs that transmit or receive MPLS-encapsulated frames. By default, MPLS and MPLS label distribution protocols are disabled when MPLS is first configured on a VLAN.

Enabling LDP on the Switch

- To enable LDP on the switch, use the following command:

  ```
  enable mpls protocol ldp
  ```

By default, LDP is disabled on the switch. This command enables *MPLS* to process LDP control packets and to advertise and receive LDP labels.

Note
Globally enabling the LDP protocol does not enable LDP on MPLS configured *VLAN*s. See the next section for instructions on enabling LDP on a VLAN.

Enabling and Disabling LDP on a VLAN

Each *VLAN* must be specifically LDP-enabled in order to set up LDP adjacencies over the interface. Before LDP can be enabled on a VLAN, the VLAN must be configured to support *MPLS* as described in Adding MPLS Support to VLANs on page 1107.

- To enable LDP on a VLAN, use the following command:

  ```
  enable mpls ldp [{vlan} vlan_name | vlan all]
  ```

  This command enables LDP on one or all VLANs for which MPLS has been configured.

  Note
  If you have enabled LDP and MPLS on an IGP interface that is used to reach a particular destination, make sure that you enable LDP and MPLS on all additional IGP interfaces that can reach that same destination (for example, enable LDP and MPLS on all *OSPF* VLANs that are connected to the backbone network).

- To disable LDP on a VLAN, use the following command:

  ```
  disable mpls ldp [{vlan} vlan_name | vlan all]
  ```

  This command disables LDP on one or all VLANs for which MPLS has been configured. This command terminates all LDP hello adjacencies and all established LDP LSPs that use the specified interface(s).

Creating Static LSPs

Static LSPs are label switched paths that are manually configured at each LSR in the prospective path. Static LSPs are too labor intensive for building complex network topologies, but they can be useful for defining simple one-hop LSPs to MTUs that might not have the CPU power necessary to support routing and label distribution protocols.

Figure 135 shows two static LSPs configured between VLAN A and VLAN B. The dashed line shows the static LSP for unidirectional communications from VLAN A to VLAN B. The solid line shows the static LSP for communications in the reverse direction.

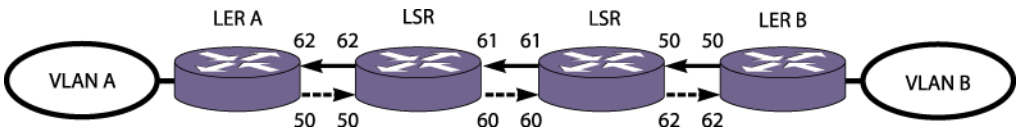


Figure 135: Static LSP Example

The path that an LSP takes and the labels the LSP uses at every hop do not change when the network topology changes due to links and nodes going up or down. Once enabled, a static LSP remains administratively up until it is manually disabled.

Static LSP configuration is different for ingress, transit, and egress LSRs. At the ingress LER, only an egress label is defined. At transit LSRs, both ingress and egress labels must be defined, and at the egress LER, only the ingress label is defined. During configuration, you must ensure that the egress label number for each LSR matches the ingress label number for the downstream LSR.

When creating static LSPs, consider the following guidelines:

- It is your responsibility to ensure that the egress label of an LSR matches the ingress label of the downstream LSR. The software does not detect or report label mismatches. Mismatches result in dropped or mis-routed packets.
- The operational state of the LSP is set to down at the head-end on local failures. However, there is no mechanism to detect the LSP going down when a failure occurs on a downstream node. When a failure occurs at a downstream node, traffic may be black-holed for the duration of the failure.
- The traffic profile for static LSPs is not configurable in this release. All static LSPs are given best effort treatment.
- The maximum number of static LSPs configurable on any given node is 1024. The maximum number of ingress static LSPs that are used to forward traffic to a single destination is 16 (as limited by *ECMP*).
- When multiple LSPs exist for the same destination, unless forced otherwise, signaled LSPs are preferred to static LSPs. When choosing an LSP for a FEC, the software prefers RSVP-TE LSPs first, LDP next, and finally static LSPs.
- Since the software has no knowledge of the cost or hop-count associated with each static LSP, all static LSPs to the same destination are equally preferred by IP routing.
- We recommend that the same LSP name be used on every LSR along the path of the static LSP. The software does not check for naming consistency across LSRs. However the switch does report an error when the configured name is not unique to the LSP on that LSR.

To configure a static LSP, use the following procedure at each node on the path:

1. Create a namespace for the LSP using the following command:

   ```
   create mpls static lsp lsp_name destination ipaddress
   ```

2. Configure the appropriate labels for the LSP using the following command:

   ```
   configure mpls static lsp lsp_name [{egress [egress_label |
   implicit-null] egress-vlan evlan_name next-hop ipaddress} {ingress
   ingress_label {ingress-vlan ivlan_name}}]
   ```

3. Configure optional traffic restrictions for IP or VPN traffic as needed using the following command:

   `configure mpls static lsp lsp_name transport [ip-traffic [allow | deny] | vpn-traffic [allow {all | assigned-only} | deny]]`

4. Enable the static LSP for operation using the following command:

   `enable mpls static lsp {lsp_name | all }`

5. When the configuration is complete, you can view the static LSP configuration with the following command:

   `show mpls static lsp {summary | {lsp_name} {detail}}`

6. Clear the counters for the static LSP using the command:

   `clear counters mpls static lsp {lsp_name | all }`

7. Once the static LSP is created on all path nodes, you can configure a default route, an IP route, or a VPN route to use the LSP. To configure a default or IP route to use the LSP, use the following command:

   `configure iproute add default [{gateway {metric} {vr vr_name} {unicast-only | multicast-only}} | {lsp lsp_name {metric}}]`

8. To configure a VPN route to use the LSP, use the following command:

   `configure l2vpn [vpls vpls_name | vpws vpws_name] peer ipaddress [add | delete] mpls lsp lsp_name`

9. To disable a static LSP, use the following command:

   `disable mpls static lsp {lsp_name | all }`

10. To delete a static LSP, use the following command:

    `delete mpls static lsp [lsp_name | all]`

Configuring Penultimate Hop Popping

- To enable or disable PHP, use the following command:

  `enable mpls php [{vlan} vlan_name | vlan all]`

  This command enables or disables whether PHP is requested by the egress LER. If vlan all is selected, PHP is enabled on all VLANs on which *MPLS* has been configured. By default, PHP is disabled.

  When PHP is enabled, PHP is requested on all LSPs for which the switch is the egress LER.

  PHP is requested by assigning the Implicit Null Label in an advertised mapping. PHP is always performed when requested by a peer (for example, when acting as an intermediate LSR).

Configuring QoS Mappings

The ExtremeXOS software provides examination and replacement services for the EXP field.

These services behave like the dot1p *QoS* commands. When EXP examination is enabled, the EXP value from the received frame is used to assign the packet to a QoS

profile. Once a packet is assigned to a QoS profile, the EXP value may be overwritten in hardware before being transmitted. By default, the QoS profile EXP value is equivalent to the QoS profile number one (QP1). To enable QoS for *MPLS* LSPs, use the following command: `enable mpls exp examination`

This command enables EXP examination for MPLS received packets. When EXP examination is enabled, the EXP field in the outer or top label of the label stack is used to assign the received packet to an internal switch qosprofile. If enabled, all MPLS packets are mapped to a qosprofile based on the configured EXP bit value. That is, a packet with an EXP value of 0 is mapped to qosprofile QP1, a packet with an EXP value of 1 is mapped to qosprofile QP2, and so on. By default, EXP examination is disabled and all received MPLS packets are sent to QP1.

Each EXP value can be assigned a qosprofile. Multiple EXP values can be assigned to the same qosprofile. The following command is used to configure the switch to route packets with a received EXP value to a specific qosprofile:

`configure mpls exp examination {`**`value`**`} `*`value`*` {`**`qosprofile`**`} `*`qosprofile`*

The switch can overwrite the EXP value in the outer label of an MPLS packet.

The following command is used to enable the switch to replace the EXP value:

`enable mpls exp replacement`

This command enables EXP replacement for MPLS transmitted packets. When EXP replacement is enabled, the EXP field in the outer or top label of the label stack is replaced with the EXP value configured for the QoS profile for which the packet was assigned. By default, EXP replacement is disabled and packets are propagated without modifying the EXP field value.

Each qosprofile can be assigned an EXP value used to overwrite the EXP bit field in the outer label of the transmitted packet.

All qosprofiles can be configured to overwrite the EXP bit field using the same EXP value. The following command is used to configure the switch to overwrite MPLS packets transmitted from a specific qosprofile with a new EXP value:

`configure mpls exp replacement {`**`qosprofile`**`} `*`qosprofile`*` {`**`value`**`} `*`value`*

Mapping Dot1p to EXP Bits

The priority of Ethernet tagged packets can be mapped into the *MPLS* network and vice versa using the switch fabric qosprofile.

Ethernet packets are assigned to a qosprofile by enabling dot1p examination. MPLS packets are assigned to a qosprofile by enabling exp examination. When the packets egress the switch, the dot1p and exp bit fields can be overwritten.

Enabling EXP replacement instructs the switch to overwrite both the dot1p field and the exp field in the outer most label.

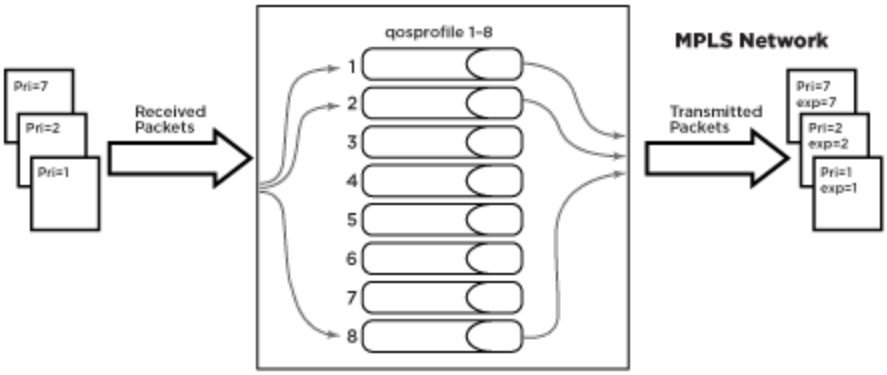


Figure 136: Mapping Dot1p to EXP Bits

By default, when dot1p examination and exp replacement are not enabled, all received packets are routed through qosprofile qp1 and the packets are transmitted with the dot1p and exp value of zero. As shown in Figure 136, the switch can be configured to route packets to a specific qosprofile based on the dot1p value in the 802.1p bit field. In this example dot1p examination is enabled. By default, when dot1p examination is enabled, packets are sent to the qosprofile that corresponds dot1p type to qosprofile mapping.

This mapping can be viewed using `show dot1p type`. By default, MPLS exp replacement is disabled. By enabling MPLS exp replacement, MPLS packets are transmitted with the configured qosprofile dot1p and exp value. By default, these values correspond to the qosprofile. Qosprofile 1 overwrites the dot1p and exp fields with 0, qosprofile 2 overwrites the dot1p and exp fields with 1, and so on.

To configure the reverse packet flow, mpls exp examination and dot1p replacement must be configured. Enabling MPLS exp examination instructs the switch to route received MPLS packets to a qosprofile based on the EXP value. Enabling dot1p replacement instructs the switch to write the dot1p value in the transmitted packet.

Enabling and Disabling LDP Loop Detection

There are two types of LDP loop detection. By default both are disabled. Enabling loop detection allows the switch to detect if a label advertisement loop exists within the network. Both hop count and path vector loop detection methods are used if loop detection is enabled.

- To enable LDP loop detection, use the following command:

  ```
  enable mpls ldp loop-detection
  ```

- To disable LDP loop detection, use the following command:

  ```
  disable mpls ldp loop-detection
  ```

These commands affect the LDP loop detection behavior for all LDP enabled VLANs.

Configuring an LDP Label Advertisement Filter

- To configure an LDP label advertisement filter, use the following command:

  ```
  configure mpls ldp advertise [{direct [all | lsr-id | none]} | {rip
  [all | none] | {static [all | none]}
  ```

  This command configures a filter to be used by LDP when originating unsolicited label mapping advertisements to LDP neighbors.

  You can configure how the direct advertisement filter is applied, as follows:

  - direct—The advertisement filter is applied to the FECs associated with direct routes.
  - rip—The advertisement filter is applied to the FECs associated with *RIP* routes.
  - static—The advertisement filter is applied to the FECs associated with static routes.

  You can configure the advertisement filter, as follows:

  - all—Unsolicited label mappings are originated for all routes of the specified type (direct, RIP, or static).
  - lsr-id—An unsolicited label mapping is originated if a /32 direct route exists that matches the *MPLS* LSR ID. This filter is the default setting for direct routes and is only available for direct routes.
  - none—No unsolicited label mappings are originated for all routes of the specified type. This is the default setting for RIP and static routes.

  You can control the number of labels advertised using the configure mpls ldp advertise command. Advertising labels for a large number of routes can increase the required number of labels that must be allocated by LSRs. Care should be used to insure that the number of labels advertised by LERs does not overwhelm the label capacity of the LSRs.

Configuring LDP Session Timers

- To configure LDP session timers, use the following command:

  ```
  configure mpls ldp timers [targeted | link] [{hello-time
  hello_hold_seconds} {keep-alive-time keep_alive_hold_seconds}]
  ```

  This command configures the LDP peer session timers for the switch. The LDP peer session timers are separately configurable for link and targeted LDP hello adjacencies.

  The hello-time *hello_hold_seconds* parameter specifies the amount of time (in seconds) that a hello message received from a neighboring LSR remains valid. The rate at which hello messages are sent is one third the configured hello-time. If a hello message is not received from a particular neighboring LSR within the specified hello-time *hello_hold_seconds* then the hello-adjacency is not maintained with that neighboring LSR. Should two peers have different configured hello-time values, they negotiate to use the lower value.

The session keep-alive time *keep_alive_hold_seconds* parameter specifies the time (in seconds) during which an LDP message must be received for the LDP session to be maintained. The rate at which keep alive messages are sent, provided there are no LDP messages transmitted, is one sixth the configured keep-alive-time. If an LDP PDU is not received within the specified session keep-alive time *keep_alive_hold_seconds* interval, the corresponding LDP session is torn down. Should two peers have different configured keep-alive-time values, they negotiate to use the lower value.

In the event that two peers have both a link and a targeted hello adjacency, the hello-time values for the two hello adjacencies are negotiated separately. The keep-alive-time value is established based on negotiations occurring when the LDP session is established following the first hello adjacency to be established.

The minimum and maximum values for both the hello-time *hello_hold_seconds* and keep-alive time *keep_alive_hold_seconds* are 6 and 65,534, respectively. Changes to targeted timers only affect newly created targeted peers. Disabling and then enabling all VPLS instances causes all current targeted peers to be re-created.

The default values are as follows:

- link ldp hello-time *hello_hold_seconds* – 15
- targeted-ldp hello-time *hello_hold_seconds* – 45
- link ldp hello-time *interval_time* – auto set to 1/3 the configured hello-time
- targeted-ldp hello-time *interval_time* – auto set to 1/3 the configured hello-time
- link ldp keep-alive *keep-alive hold-seconds* – 40
- targeted-ldp keep-alive *keep-alive hold-seconds* – 60
- link ldp keep-alive *interval_time* – auto set to 1/6 the configured keep-alive time
- targeted-ldp keep-alive *interval_time* – auto set to 1/6 the configured keep-alive time

Restoring LDP Session Timers

- To restore the default values for LDP session timers, use the following command:

  ```
  unconfigure mpls
  ```

  This command can only be executed when *MPLS* is disabled, and it restores all MPLS configuration settings.

Clearing LDP Protocol Counters

- To clear the LDP control protocol error counters, use the following command:

  ```
  clear counters mpls ldp {{{vlan} vlan_name} | lsp all}
  ```

  Omitting the optional *vlan* parameter clears counters for all LDP enabled interfaces.

Resetting MPLS Configuration Parameter Values

- To reset *MPLS* configuration parameters to their default values, disable MPLS (see Enabling and Disabling MPLS on an LSR on page 1108) and then use the following command:

  ```
  unconfigure mpls
  ```

  This command affects the following configuration parameters:

  - All VLANs are removed from MPLS.
  - All EXP (examination and replacement) QOS mappings are reset.
  - LSR-ID is reset.
  - LDP and RSVP-TE are globally disabled.
  - MPLS Traps are disabled.
  - LDP and RSVP-TE timers are reset.
  - LDP Advertisement settings are reset.
  - LDP Loop Detection settings are reset.
  - All RSVP-TE LSPs are deleted.
  - All RSVP-TE Paths are deleted.
  - All RSVP-TE Profiles are deleted.
  - The default RSVP-TE Profile is reset.

Managing the MPLS BFD Client

Bidirectional Forwarding Detection (BFD) is introduced in Bidirectional Forwarding Detection (BFD) on page 1309.

The *MPLS* BFD client enables rapid detection of failures between MPLS neighbors on specific *VLAN*s. BFD detects forwarding path failures at a uniform rate, which makes the re-convergence time consistent and predictable and makes network profiling and planning easier for network administrators.

When BFD detects a communication failure, it informs MPLS, which treats the indication as an interface (VLAN) failure. This allows the MPLS protocols to quickly begin using alternate paths to affected neighbors (the methodology for selecting alternate paths is dependent upon the MPLS protocol in use and how it reacts to interface failure conditions). As MPLS connections (LSPs) are removed from the interface, BFD sessions are removed as well, and the interface returns to a state without BFD protection. The MPLS protocol might continue to attempt to reestablish LSP connections across the interface, and if successful, also attempt to establish a BFD session with the corresponding neighbor. MPLS does not process BFD state changes until the BFD

session is fully active in the UP state, at which point state changes are processed and the state for LSPs which cross the interface becomes BFD protected.

Note
BFD sessions are established only when both peers select the same LSP route. We recommend that BFD operate only on interfaces that have one peer.

- To enable the MPLS BFD client, use the following command:

  ```
  enable mpls bfd [{vlan} vlan_name | vlan all]
  ```

  Note
  BFD must be enabled on the interface before sessions can be established. To enable BFD, use the command: `[enable | disable] bfd vlan vlan_name` .

- To disable the MPLS BFD client, use the following command:

  ```
  disable mpls bfd [vlan all | {vlan} vlan_name] {delete-sessions}
  ```

- To display the MPLS BFD client information, use the following commands:

  ```
  show mpls interface {{vlan} vlan_name} {detail}
  show mpls bfd [{vlan} vlan_name | ip_addr]
  ```

Displaying MPLS Configuration Information

Displaying MPLS Basic Configuration Information

- To display basic *MPLS* configuration information, use the following command:

  ```
  show mpls
  ```

  This command displays the general configuration of all the MPLS components and system wide configuration.

The output, shown below, displays the switch MPLS, RSVP-TE, and LDP configuration status. It also shows the configuration status of *SNMP* trap, EXP examination, and EXP replacement settings. The configured LSR ID are also shown.

```
# show mpls
Virtual Router Name                    : VR-Default
MPLS Admin Status                      : Enabled
MPLS Oper Status                       : Enabled
RSVP-TE Admin Status                   : Enabled
RSVP-TE Oper Status                    : Enabled
LDP Admin Status                       : Enabled
LDP Oper Status                        : Enabled
MPLS SNMP Traps                        : Disabled
L2VPN SNMP Traps                       : Disabled
EXP Examination                        : Enabled
EXP Replacement                        : Disabled
LSR ID                                 : 192.99.1.5
```

Displaying LDP Basic Configuration Information

- To display basic LDP configuration information, use the following command:

  ```
  show mpls ldp
  ```

  This command displays the general configuration of LDP.

  Some settings are not configurable. These fields are identified with an asterisk (*). The remaining fields can be modified using LDP configuration commands to change the behavior of LDP. A list of VLANs that have LDP enabled is shown at the bottom of the display output.

```
# show mpls ldp
LDP Status                : Enabled
Protocol Version          : v1*
Label Retention Mode      : Liberal*
Label Distribution Method : Downstream Unsolicited*
LDP Loop Detection
Status            : Disabled
Hop-Count Limit   : 255
Path-Vector Limit : 255
LDP Targeted Timers
Hello Hold      : 45 seconds
Keep Alive Hold : 60 seconds
LDP Link Timers
Hello Hold      : 15 seconds
Keep Alive Hold : 40 seconds
Label Advertisement
Direct : All
Rip    : None
Static : None
LDP VLANs  : loopback
: blowingrock
: boone
: asheville
* Indicates parameters that cannot be modified
```

Displaying MPLS Interface Information

- To display the *MPLS* interface information, use the following command:

  ```
  show mpls interface {{vlan} vlan_name} {detail}
  ```

  When the optional parameters are omitted, this command displays information for all the configured MPLS *VLAN* interfaces.

  The summary MPLS interface information displayed includes the configured IP address, approximate time RSVP-TE and LDP have been up, number of neighbors or adjacencies, and a set of status flags.

  When the *vlan_name* parameter is specified, this command displays the current MPLS interface summary configuration and status for only the specified VLAN. If the optional detail keyword is specified, the summary information is displayed in the detail format.

Displaying LDP Interface Information

- To display LDP interface information, use the following command:
  `show mpls ldp interface {{vlan} vlan_name} {detail | counters}`

  This command displays the operational LDP interface information.

  All the VLANs that have LDP enabled are listed. The negotiated LDP hello hold time is displayed. This is not the configured LDP hello hold time but represents the value that is negotiated between the local switch and the connected LDP peer. Hello messages are transmitted every 1/3 the negotiated hello hold time. In the following example, that would be every 5 seconds. The hello timer status information is displayed in milliseconds. The approximate LDP uptime and status flags are also shown.

```
# show mpls ldp interface
VLAN Name         #Adj  NegHHldTm  NxtHello  UpTm  Flags
---------------  ----  ---------  --------  ----  -----
loopback             0      15000      2230   16h  MLU
blowingrock          1      15000      2200   14h  MLU
boone                1      15000      2210   16h  MLU
asheville            1      15000      2210   16h  MLU
Flags: (M) MPLS Enabled, (L) LDP Enabled,
(U) LDP Operational
```

Displaying MPLS Label Information

- To display *MPLS* label information, use the following commands:
  `show mpls {rsvp-te | static} label {summary | label_num | [advertised | received] {label_num} | received implicit-null}`

  `show mpls label l3vpn {summary | label_num | [advertised | received] {label_num}}`

  `show mpls {ldp} label {lsp} {summary | label_num | [advertised | received] {label_num} | received implicit-null}`

  `show mpls ldp label {lsp} advertised implicit-null {ipNetmask}`

  `show mpls {ldp} label l2vpn {summary | label_num | [advertised | received] {label_num}}`

  `show mpls ldp label l2vpn retained {ipaddress}`

  `show mpls ldp label lsp retained {ipNetmask}`

  `show mpls ldp label retained [l2vpn {ipaddress} | lsp {ipNetmask}]`

Displaying MPLS Label Mapping Information

- To display *MPLS* label mapping information, use the following command:
  `show mpls ldp lsp {prefix ipNetmask} {ingress | egress | transit} {detail}`

  `show mpls rsvp-te lsp [egress | transit] {fast-reroute} {{lsp_name} {[destination | origin] ipaddress} {detail} | summary}`

  This command displays information about how to forward packets that arrive labeled as MPLS packets. As such, it shows how labels advertised to upstream peers

are mapped to labels received from downstream peers. This mapping is sometimes referred to as the Incoming Label Map (ILM).

When the label_number parameter is omitted, summary information is displayed for all incoming label assignments that have been made by the switch. When the label_number is specified, summary information is displayed for the specified label.

As can be seen below, the output display differs for label mappings signaled by LDP and RSVP-TE. Please see the respective sections for additional information.

```
# show mpls ldp lsp
Prefix            Adv Label  Peer Label  Next Hop        VLAN
192.168.0.4/32     0x80402    --          --               lpbk
192.168.0.2/32     0x00039    0x80400     11.0.2.2         vlan2
11.0.4.0/24        0x0003A    0x80401     11.0.2.2         vlan2
192.168.0.4/32     0x0003B    0x00013     11.0.2.2         vlan2
* BD-10K.6 # show mpls rsvp-te lsp
Ingress LSP Name Path Name        Destination     Transmit I/F     UpTm
Flags
---------------- ---------------- --------------- ---------------- ----
--------
LSR1-LSR4        path1-2-4        192.168.0.4     vlan2             47m
UEP--OIV
Egress LSP Name  Source IP       Destination     Receive I/F      UpTm
---------------- --------------- --------------- ---------------- ----
LSR4-LSR1        192.168.0.4     192.168.0.1     vlan1             47m
Transit LSP Name Source IP       Destination     Receive I/F   Transmit I/F  UpTm
---------------- --------------- --------------- ------------- ------------- ----
LSR2-LSR3        192.168.0.2     192.168.0.3     vlan2          vlan1        47m
Flags: (U) Up, (E) Enabled, (P) Primary LSP, (S) Secondary LSP,
(R) Redundant Paths, (B) Bandwidth Requested, (O) ERO Specified,
(I) IP Traffic Allowed, (V) VPN Traffic Allowed,
(v) VPN Assigned Traffic Allowed
```

Displaying MPLS QoS Mapping Information

- To display *MPLS QoS* mapping information, use the following command:

```
* BD-10K.10 # show mpls exp examination
EXP --> QoS Profile mapping:
00 --> QP1
01 --> QP2
02 --> QP3
03 --> QP4
04 --> QP5
05 --> QP6
06 --> QP7
07 --> QP8
EXP Examination is disabled
* BD-10K.11 # show mpls exp replacement
QoS Profile --> EXP mapping:
QP1 --> 00
QP2 --> 01
QP3 --> 02
QP4 --> 03
QP5 --> 04
QP6 --> 05
QP7 --> 06
QP8 --> 07
EXP Replacement is disabled
* BD-10K.12 #
```

Configured mappings for both dot1p-to-exp and exp-to-dot1p are displayed.

Displaying LDP Peer Session Information

- To display *MPLS* LDP peer session information, use the following command:
  `show mpls ldp peer {ipaddress} {detail}`

  This command displays information about the status of LDP peer sessions. Summary information is displayed for all known LDP peers and LDP peer sessions. If you specify the *ipaddress* of an LDP peer, information for the single LDP peer is displayed. If you specify the detail keyword, additional information is displayed in a comprehensive detailed format.

  By default the information displayed includes:

  - Peer sessions
  - Peer state
  - Uptime
  - Number of hello adjacencies

  If you specify the detail keyword, the following additional information is displayed:

  - Discontinuity time
  - Negotiated label distribution
  - Next hop address
  - Keep-Alive hold timer
  - Hello adjacency details

Displaying LDP Protocol Counters

LDP control protocol packet error counters are maintained per interface.

- To view these counters, use the following command:
  `show mpls ldp interface {{vlan} vlan_name} {detail | counters}`

  These counters may be useful in determining LDP issues between peers.

  Error counters that continually increment should be investigated.

```
# show mpls ldp interface vlan blowingrock counters
VLAN: blowingrock (192.60.40.5)
Link     Targeted
Counter                          Adjacencies  Adjacencies
-------------------------------  -----------  -----------
Shutdown Notifications (Rcvd)              0            0
Shutdown Notifications (Sent)              0            0
Failed Session Attempts (NAKs)             0            0
Hello Errors                               0            0
Parameters Advertised Errors               0            0
Max PDU Length Errors                      0            0
Label Range Errors                         0            0
Bad LDP ID Errors                          0            0
Bad PDU Length Errors                      0            0
Bad Msg Length Errors                      0            0
Bad TLV Length Errors                      0            0
```

```
Bad TLV Value Errors                    0            0
Keep-Alive Timeout Errors               0            0
```

Omitting the optional vlan parameter displays counters for all LDP enabled interfaces.

Displaying LDP LSP Forwarding Database

- To display information about LDP LSPs, use the following command:
  `show mpls ldp lsp {**prefix** *ipNetmask*} {**egress** | **ingress** | **transit**} {**detail**}`

  This command displays the LDP LSPs established to, from, and through this switch. By default, ingress, egress, and transit LSPs are all displayed. By optionally specifying the LSP type, the output display is filtered to show only the type of LSPs specified.

  When all LSP types are being displayed, LSPs that show only an advertised label represent egress LSPs from the network perspective or incoming LSPs from the switch perspective. LSPs that show only a received label represent ingress LSPs from the network perspective or outgoing LSPs from the switch perspective. LSPs that show both an incoming and an outgoing label represent transit LSPs. As Extreme switches are merge-capable, all LDP transit LSPs can also be used as ingress LSPs.

  The significance of the *VLAN* information shown depends on the LSP type. For ingress and transit LSPs, the indicated VLAN is the *MPLS* interface used to reach the next hop peer. For egress LSPs, there is no associated MPLS next hop interface. When the prefix being advertised is associated with a local (direct) VLAN, that VLAN name is displayed. When the advertised prefix is associated with a static or an *RIP* route, the VLAN field is empty.

  Advertised labels have switch-wide significance and are generally advertised out multiple interfaces.

```
* BD-10K.15 # show mpls ldp lsp
Prefix            Adv Label  Peer Label  Next Hop        VLAN
192.99.1.5/32      0x80402    --          --              loopback
192.80.40.0/24     0x80403    --          --              asheville
192.100.40.0/24    0x80404    --          --              boone
192.60.40.0/24     0x8040b    --          --              blowingrock
192.24.100.0/24    0x00015    0x80401     192.100.40.3    boone
192.99.1.3/32      0x00016    0x80403     192.100.40.3    boone
192.10.50.0/24     0x00018    0x80405     192.100.40.3    boone
10.20.30.0/24      0x0001c    0x00013     192.100.40.3    boone
11.136.96.0/24     0x0001d    0x00014     192.100.40.3    boone
```

  Specifying the optional detail keyword displays each LSP in detail format.

  Additionally, received packets and bytes are maintained for transit and egress (or incoming) LSPs. Specifying the keyword prefix and a matching ipNetmask restricts the display to a single entry.

```
*BD-X8.17 # show mpls ldp lsp prefix 11.108.96.0/24 detail
FEC IP/Prefix: 11.108.96.0/24:
Advertised Label: 0 (0)
Received Label  : 0x18 (24)
Next Hop IP     : 192.100.40.3
```

```
VLAN Name       : boone
Packets received  :        489
Bytes received    :      46944
```

Displaying RSVP-TE LSP Configuration Information

- Tp display RSVP-TE LSP configuration information, use the following command:

  ```
  show mpls rsvp-te
  ```

  This command displays summary configuration and status information for RSVP-TE. Global status of RSVP-TE and the configured standard and rapid-retry LSP timer values are included in the display output.

Displaying the RSVP-TE Paths

- To display RSVP-TE paths, use the following command:

  ```
  show mpls rsvp-te path {path_name} {detail}
  ```

  This command displays the configuration and status information for *MPLS* RSVP-TE paths. Information is listed in tabular format and includes the path name, number of configured ERO objects, number of LSPs configured to use this path, the list of EROs and their type. Optionally specifying the detail keyword displays the path information in verbose format, including all LSPs that are configured to use the path.

Displaying the RSVP-TE Path Profile

- To display RSVP-TE path profiles, use the following command:

  ```
  show mpls rsvp-te profile {profile_name} {detail}
  ```

  By default, this command displays all configured profile parameters for the specified profile. If the profile name is omitted, the profile parameter values for all configured LSP profiles are displayed. Optionally specifying the keyword detail displays the profile information is verbose format. When the detail keyword is specified, all LSPs that are configured to use this profile are also displayed.

Displaying the RSVP-TE LSP

- To display the RSVP-TE LSP, use the following command:

  ```
  show mpls rsvp-te lsp {{[destination | origin] ipaddress} {fast-reroute} {detail} | summary}
  ```

  This command displays the LSP information associated with RSVP-TE that is used to forward packets within the *MPLS* network. If no options are specified, summary information for all RSVP-TE LSPs is displayed. This information includes the LSP name, LSP direction relative to the MPLS network (i.e., ingress, egress, or transit), incoming and or outgoing interface, configured destination, and uptime. If the optional LSP name parameter is specified, only the LSP information for the specified ingress LSP name is displayed. Optionally, the LSPs displayed can be further

qualified by the keywords ingress, egress, and transit. These keywords qualify the LSPs displayed from the perspective of the switch.

Ingress LSPs identify LSPs that originate from the switch into the MPLS network. Egress LSPs identify LSPs that terminate at the switch from the MPLS network. Transit LSPs represent LSPs that traverse the switch.

The optional destination keyword limits the display to only those LSPs that terminate at the specified IP address. The optional origin keyword limits the display to only those LSPs that originate at the specified IP address. Optionally specifying the detail keyword displays the LSP information in verbose format. Additional information displayed includes the configured path and profile, transmit and/or receive labels, failure and retry counts, packet and byte counters, and recorded routes.

MPLS Configuration Example

The network configuration, shown in Figure 137, illustrates how to configure a BlackDiamond switch to support a routed MPLS network.

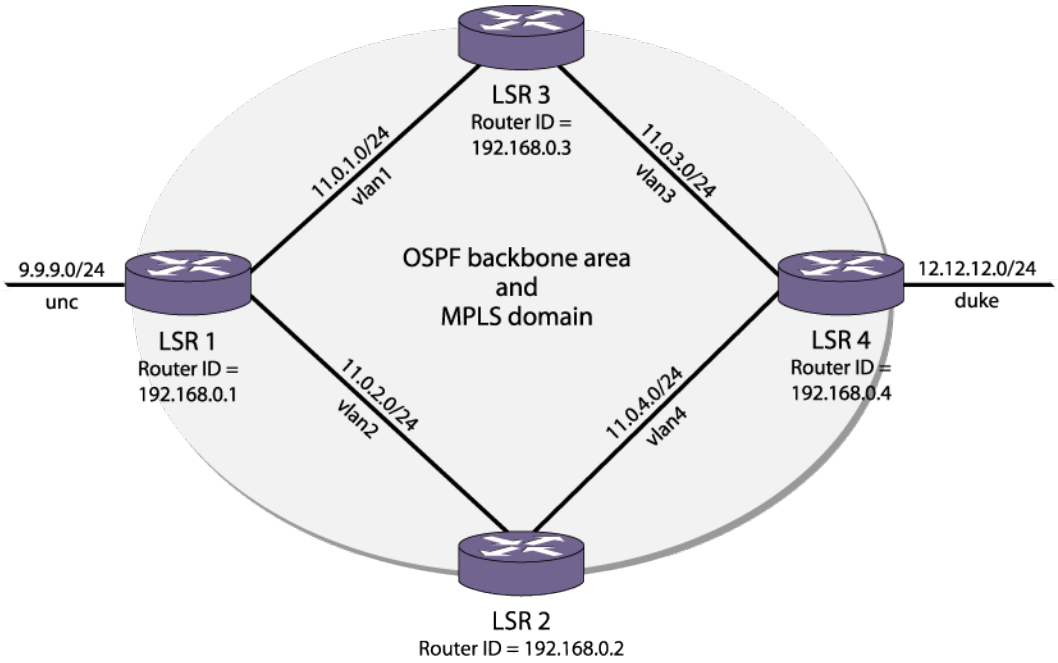


Figure 137: MPLS Configuration Example

The four switches, labeled LSR 1, LSR 2, LSR 3, and LSR 4, have the same physical hardware configuration. Each switch contains two 8900-G48T-xl and an 8900-MSM128 module. The switches are all interconnected via Gigabit Ethernet to form the *OSPF* backbone area and the MPLS domain. In this example, two directly connected OSPF-disabled VLANs are shown: unc and duke. Traffic between unc and duke follows routed paths over calculated LSPs established between LSR 1 and LSR 4.

The commands used to configure LSR 1 are described below. The remaining LSRs are configured similarly.

The following commands configure the module types for specific slots:

```
configure slot 2 module 8900-G48T-xl
configure slot 2 module 8900-G48T-xl
```

The following command sets the maximum jumbo frame size for the switch chassis to 1600:

```
configure jumbo-frame-size size 1600
enable jumbo-frame ports all
```

The following commands create the VLANs:

```
create vlan lpbk
create vlan vlan1
create vlan vlan2
create vlan unc
```

The following commands configure the VLAN IP address and assign ports participating in each VLAN:

```
configure vlan lpbk ipaddress 192.168.0.1/32
enable ipforwarding lpbk
enable loopback-mode lpbk
configure vlan default delete ports all
configure vlan vlan1 ipaddress 11.0.1.1/24
configure vlan vlan1 add port 3:2 untagged
configure vlan vlan2 ipaddress 11.0.2.1/24
configure vlan vlan2 add port 3:3 untagged
configure vlan unc ipaddress 9.9.9.1/24
configure vlan unc add port 3:4 untagged
```

The following commands enable IP forwarding on the configured VLANs.

The MTU size is increased on the MPLS VLANs to accommodate the MPLS shim header:

```
enable ipforwarding vlan vlan1
configure ip-mtu 1550 vlan vlan1
enable ipforwarding vlan vlan2
configure ip-mtu 1550 vlan vlan2
enable ipforwarding vlan unc
```

The following command configures the MPLS LSR ID:

```
configure mpls lsr-id 192.168.0.1
```

The following commands add MPLS support to VLANs lpbk, vlan1, and vlan2:

```
configure mpls add vlan lpbk
configure mpls add vlan vlan1
configure mpls add vlan vlan2
```

The following commands enable MPLS on VLANs lpbk, vlan1, and vlan2 and LDP on VLANs vlan1 and vlan2:

```
enable mpls lpbk
enable mpls vlan1
enable mpls vlan2
enable mpls ldp vlan1
enable mpls ldp vlan2
```

The following command allows LDP to advertise a label mapping for the LSR ID:

```
configure mpls ldp advertise direct lsr-id
```

The following commands globally enable LDP and MPLS on the switch:

```
enable mpls protocol ldp
enable mpls
```

The following commands add lpbk, vlan1, and vlan2 to the backbone area.

The 0.0.0.0 (backbone) area does not need to be created because it exists by default:

```
configure ospf add vlan lpbk area 0.0.0.0
configure ospf add vlan vlan2 area 0.0.0.0
configure ospf add vlan vlan1 area 0.0.0.0
```

The following command enables distribution of local (direct) interfaces into the OSPF area:

```
enable ospf export direct cost 10 type ase-type-1
```

The following command configures the OSPF router ID on the switch and enables the distribution of a route for the OSPF router ID in the router LSA.

Originating the router ID as a host route allows other routers in the same OSPF area to establish calculated LSPs for external routes to this router:

```
configure ospf routerid 192.168.0.1
```

It also allows this router to be a peer in a L2 VPN.

The following command enables OSPF:

```
enable ospf
```

Configuring MPLS Layer-2 VPNs (VPLS and VPWS)

Note
ELRP should not be used to protect VPLS service VLANs because ELRP is not aware of VPLS pseudowires.

Configuring MPLS for Establishing Layer 2 VPN Instances

As described in Using LDP to Signal PW Label Mappings on page 1071, *MPLS* and LDP must be properly configured in order to establish the PWs that make up a Layer 2 VPN.

MPLS should be enabled using the enable mpls command and LDP should be enabled using the enable mpls protocol ldp command.

Since the MPLS LSR ID is used as the local endpoint for PWs, it is highly desirable to create a loopback *VLAN* whose associated IP address is that of the MPLS LSR ID. The configured prefix length should be 32. As described in Establishing LDP LSPs to PW Endpoints on page 1070, configuring this loopback VLAN for MPLS causes the address to be included in LDP address messages. Use the `configure mpls add vlan`

vlan_name command for this. It is not required that LDP or MPLS be enabled on the VLAN for the address to be advertised by LDP. Use the configure mpls ldp advertise direct lsr-id command to initiate an LDP label mapping for an LDP LSP to the local endpoint.

Creating or Deleting a Layer 2 VPN Domain

- To create a Layer 2 VPN, use the following command:
  `create l2vpn [vpls vpls_name | vpws vpws_name] fec-id-type pseudo-wire pwid`

  This command creates a named Layer 2 VPN instance. Multiple domains can be created, each representing a different L2 VPN. The pwid is a number that is used to signal and identify which Layer 2 VPN is associated with each pseudowire. All of the pseudowires carrying traffic for a specific Layer 2 VPN are signaled with the same pwid. No Layer 2 VPN traffic is forwarded over the Layer 2 VPN until at least one peer is added to the Layer 2 VPN and a service is associated with the Layer 2 VPN. The configured pwid also doubles as the Layer 2 VPN ID.
- To delete a Layer 2 VPN, use the following command:
  `delete l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]]`

  This command deletes the named Layer 2 VPN instance or all Layer 2 VPN instances, depending on the keyword. All Layer 2 VPN peers and services associated with the deleted Layer 2 VPN instance(s) are also deleted.

Enabling or Disabling a Layer 2 VPN Domain

- To enable a Layer 2 VPN, use the following command:
  `enable l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]]`

  This command enables a named Layer 2 VPN instance. By default, a newly created Layer 2 VPN is enabled.
- To disable a Layer 2 VPN, use the following command:
  `disable l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]]`

  This command disables the named Layer 2 VPN instance. When a Layer 2 VPN is disabled, no traffic flows across the Layer 2 VPN. The pseudowires connecting this peer to all other configured peers are also terminated, so the remote peers no longer see this LSR as an active peer.

Adding or Deleting a Layer 2 VPN Peer

- To add a peer to the Layer 2 VPN, use the following command:
  `configure l2vpn [vpls vpls_name | vpws vpws_name] add peer ipaddress {{core {full-mesh | primary | secondary} | spoke}`

  For each new peer added, a pseudowire is signaled to carry traffic for this Layer 2 VPN. Up to 64 peers can be added to a VPLS; and only one peer can be added to a VPWS. For each peer added, that remote peer must also configure this local LSR as a

peer for the Layer 2 VPN. For VPLS configurations, this insures that the VPLS core is configured as a full mesh of VPLS peers.

The Layer 2 VPN names on each peer do not have to match since the pseudowire ID is used to define the Layer 2 VPN to which each pseudowire is associated.

- Delete a peer from the Layer 2 VPN, use the following command:
  `configure l2vpn [vpls vpls_name | vpws vpws_name] delete peer [ipaddress | all]`

  Once the peer is deleted, that specified peer is no longer a member of the Layer 2 VPN. For VPLS configurations, the peer must also be removed from all other VPLS peers to insure a proper full mesh and to prevent connectivity issues.

Add or Delete a Layer 2 VPN Service

- To add a service to a Layer 2 VPN, use the following command:
  `configure l2vpn [vpls vpls_name | vpws vpws_name] add service [{vlan} vlan_name | {vman} vman_name]`

  Only one service can be added to each Layer 2 VPN. Traffic associated with the service is transported over the Layer 2 VPN. Three basic types of services are supported: *VLAN*, VMAN, and port. Both the VLAN and VMAN services are specified by adding the VLAN or VMAN name to the Layer 2 VPN. The port service is configured by adding a VMAN name to the Layer 2 VPN, configuring the Layer 2 VPN to strip the VMAN tag, and adding the port as untagged to the VMAN. This allows incoming service traffic to be transported across the Layer 2 VPN exactly as it was received. See Managing Layer 2 VPN Packet Forwarding Options on page 1129 for information about configuring a Layer 2 VPN to strip the VMAN tag.
- To delete a service from a Layer 2 VPN, use the following command:
  `configure l2vpn [vpls vpls_name | vpws vpws_name] delete service [{vlan} vlan_name | {vman} vman_name]`

  Since there is no local service that needs to be connected to the Layer 2 VPN, the pseudowires to each of the configured peers for this Layer 2 VPN are terminated.

  Note
  Ports added to Layer 2 VPN service VPLS/VMAN should not be part of PVLAN or VLAN aggregation or VLAN translation.

Enabling or Disabling a Layer 2 VPN Service

- To enable a Layer 2 VPN service, use the following command:
  `enable l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]] service`

  By default, any configured Layer 2 VPN service is enabled.
- To disable a service from the Layer 2 VPN, use the following command:
  `disable l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]] service`

When the service is disabled, the service is disconnected from the Layer 2 VPN and disabled such that no packets are sent to or received from the service. The pseudowires to each of the configured peers for this Layer 2 VPN are terminated.

Managing Layer 2 VPN Packet Forwarding Options

- To configure Layer 2 VPN packet forwarding options, use the following command:

  `configure l2vpn [vpls vpls_name | vpws vpws_name] {dot1q [ethertype hex_number | tag [include | exclude]]} {mtu number}`

  The options should be configured the same for every LSR for this Layer 2 VPN in order to prevent connectivity issues. Specifying the dot1q ethertype forces the switch to overwrite the dot1q ethertype value in the service packet. This can be used to interconnect two customer segments over the Layer 2 VPN that are using different configured ethertype values. By default, the dot1q tag in the service packet is not included. The switch can be configured to strip or exclude the dot1q tag. This can be used to emulate port services or for interoperability with equipment that may require tags.
- To unconfigure Layer 2 VPN packet forwarding options, use the following command:

  `unconfigure l2vpn [vpls vpls_name | vpws vpws_name] dot1q ethertype`

  This command resets the dot1q ethertype for the specified Layer 2 VPN to the default ethertype configured for the switch.

Configuring the Layer 2 VPN MTU

The Maximum Transmission Unit (MTU) is the maximum packet size (excluding the data link header) that an interface can transmit. The Layer 2 VPN MTU defines the MTU for the customer facing interface. By default, the MTU size is set to 1500 bytes, which is the MTU size for standard Ethernet frames.

The frame size of the customer packet also includes the data link header and FCS field. The Ethernet data link header includes a minimum of a destination MAC address (six bytes), a source MAC address (six bytes), and an ethertype field (two bytes). The FCS field is four bytes. For a 1500-byte customer payload, the minimum customer frame size is 1518 bytes.

The Layer 2 frame, minus the FCS field, is encapsulated to form an *MPLS* packet to be transmitted to a Layer 2 VPN peer. This encapsulation adds another Ethernet header, an MPLS shim header, and a FCS field. The MPLS shim header usually includes two four-byte MPLS labels. Therefore, this encapsulation adds a minimum of 26 bytes.

The total frame size includes the customer payload, the Ethernet header on the customer facing interface (minimum 14 bytes), and the MPLS encapsulation (minimum 26 bytes). For a 1500-byte customer payload, the minimum Layer 2 VPN frame size is 1540 bytes.

The total frame size is restricted by the jumbo frame size. The maximum jumbo frame size that can be configured is 9216 bytes. If the customer payload in the above example

is increased to 9176 bytes, the resulting Layer 2 VPN encapsulated frame size is 9216 bytes. Therefore, the maximum practical value of the Layer 2 VPN MTU is 9176 bytes.

There are additional considerations that can lower the maximum practical value of the Layer 2 VPN MTU setting. If the Layer 2 VPN is configured to include the *VLAN* tag of the customer interface, this increases the total frame size by four bytes. If the MPLS interface used to transmit the encapsulated Layer 2 VPN packet is a tagged VLAN, this also increases the total frame size by four bytes. If either of these configuration options is present, the maximum practical Layer 2 VPN MTU size is reduced accordingly.

- To configure the Layer 2 VPN MTU size, use the **mtu** option in the following command:

  `configure l2vpn [vpls vpls_name | vpws vpws_name] {dot1q [ethertype hex_number | tag [include | exclude]]} {mtu number}`

  The Layer 2 VPN MTU value is signaled to peers as part of the VC FEC information during PW setup. The local value must match the value received from the PW peer. If the two values do not match, the PW is not established.

Managing VPLS Redundancy Options

- To configure VPLS redundancy options, use the following command:

  `configure {l2vpn} vpls vpls_name redundancy [esrp esrpDomain | eaps | stp]`
- To unconfigure VPLS redundancy options, use the following command:

  `unconfigure {l2vpn} vpls vpls_name redundancy [eaps | esrp | stp]`
- To display the configured VPLS redundancy option, use the following command:

  `show l2vpn vpls {vpls_name} {peer ipaddress} detail`

Displaying Layer 2 VPN Status

- To display the status of all Layer 2 VPNs, a configured Layer 2 VPN, or a specific peer within a specific Layer 2 VPN, use the following command:

  `show [ {l2vpn} vpls {vpls_name} | l2vpn vpws {vpws_name} | l2vpn ] {peer ipaddress} {detail} | summary }`

Displaying Layer 2 VPN Statistics

- To display statistics for Layer 2 VPNs, use the following command:

  `show mpls statistics l2vpn {vpls_name | vpws_name } {detail}`

Managing Layer 2 VPN SNMP Traps

- To enable or disable Layer 2 VPN *SNMP* traps, use the following commands:

  `enable snmp traps l2vpn`

  `disable snmp traps l2vpn`

- To configure or remove a text-string identification for traps from a Layer 2 VPN, use the following command:

  ```
  configure vpls vpls_name snmp-vpn-identifier identifier
  unconfigure vpls vpls_name snmp-vpn-identifier
  ```

- To view the configured state for Layer 2 VPN SNMP traps, use the following command:

  ```
  show mpls
  ```

VPLS VPN Configuration Examples

Basic Point-to-Point VPLS Configuration Example

This *MPLS* VPLS network configuration shown Figure 138, builds upon the routed MPLS network configuration example shown in MPLS Configuration Example on page 1124.

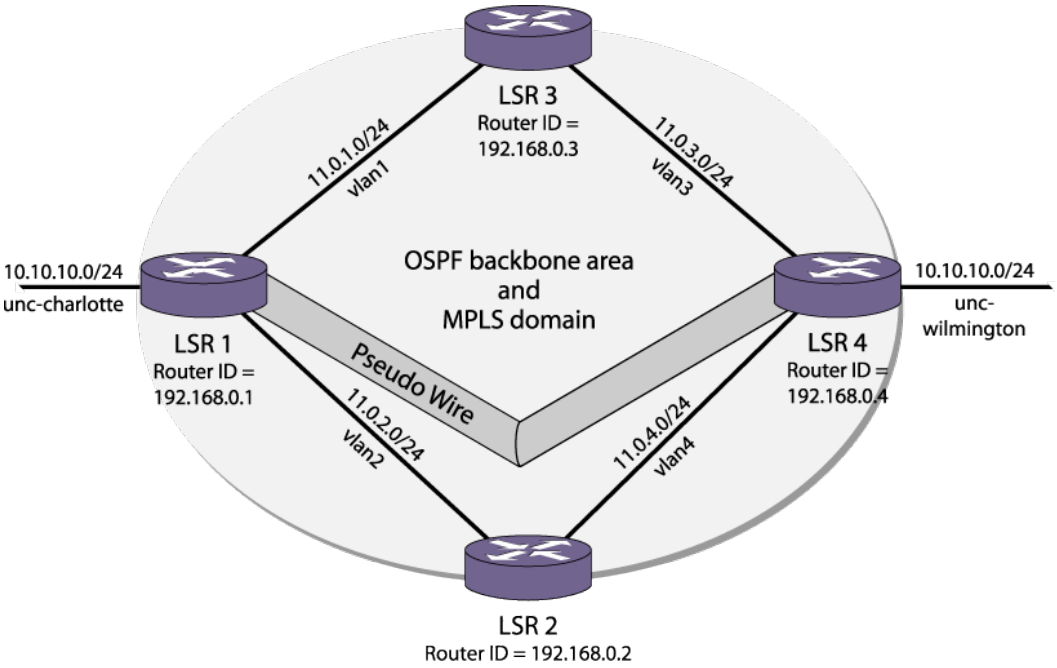


Figure 138: MPLS VPLS Configuration Example

Assuming that the routed MPLS network has already been configured as in the previous example, the commands used to create the VPLS on LSR1 and LSR4 follow. The nc-university-vpn is a point-to-point VPLS, since only two peers are participating.

The following command creates the VPLS with VPN ID 35. This command must be issued on both LSR1 and LSR4:

```
create l2vpn vpls nc-university-vpn fec-id-type pseudo-wire 35
```

The following command enables the VPLS instance specified by the *vpls_name*:

```
enable l2vpn [vpls [ vpls_name | all ]
```

On LSR1, configure the local *VLAN* service and the VPLS peer:

```
configure l2vpn vpls nc-university-vpn add service vlan unc-charlotte
configure l2vpn vpls nc-university-vpn add peer 192.168.0.4
```

On LSR4, configure the local VLAN service and the VPLS peer:

```
configure l2vpn vpls nc-university-vpn add service vlan unc-wilmington
configure l2vpn vpls nc-university-vpn add peer 192.168.0.1
```

Multipoint Full Mesh VPLS Configuration Example

The example shown in Figure 139 configures a four node full-mesh *MPLS* VPLS configuration by adding two additional peers to the previous example.

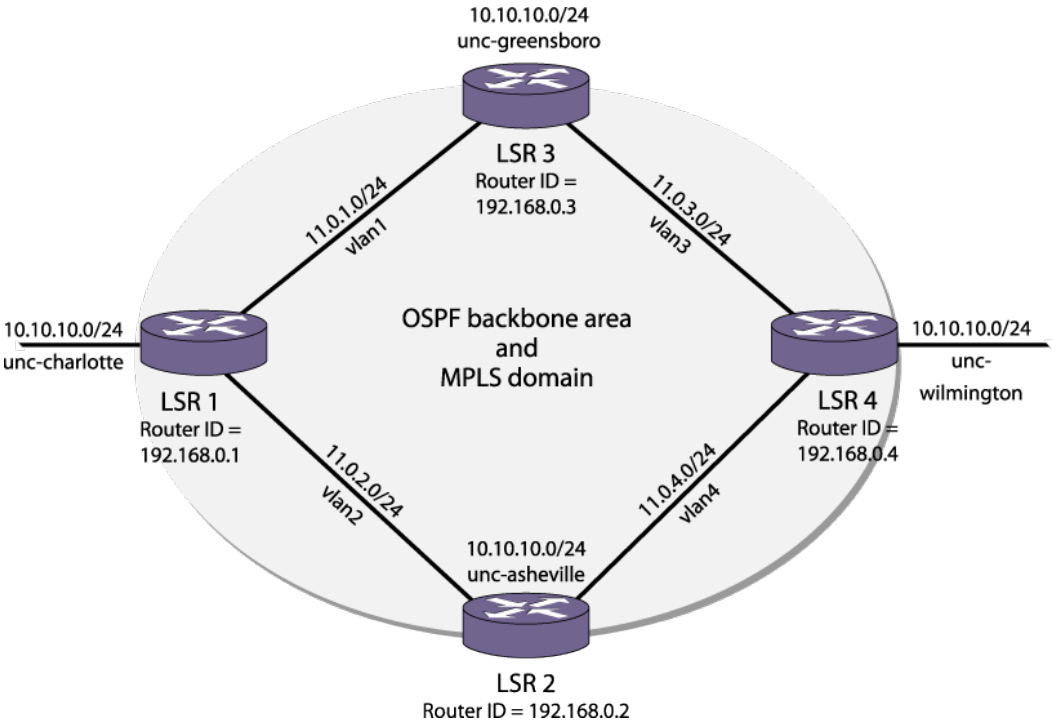


Figure 139: Full Mesh Configuration Example

LSR1

```
configure vpls nc-university-vpn add peer 192.168.0.2
configure vpls nc-university-vpn add peer 192.168.0.3
```

LSR2

```
create vpls nc-university-vpn fec-id-type pseudo-wire 35
configure vpls nc-university-vpn add peer 192.168.0.1
configure vpls nc-university-vpn add peer 192.168.0.3
configure vpls nc-university-vpn add peer 192.168.0.4
configure vpls nc-university-vpn add service vlan unc-asheville
```

LSR3

```
create vpls nc-university-vpn fec-id-type pseudo-wire 35
configure vpls nc-university-vpn add peer 192.168.0.1
```

```
configure vpls nc-university-vpn add peer 192.168.0.2
configure vpls nc-university-vpn add peer 192.168.0.4
configure vpls nc-university-vpn add service vlan unc-greensboro
```

LSR4

```
configure vpls nc-university-vpn add peer 192.168.0.2
configure vpls nc-university-vpn add peer 192.168.0.3
```

VPLS with Redundant EAPS Configuration Example

The following sections provide examples of how to configure switches to support the configuration described in VPLS EAPS Redundancy Overview on page 1086:

- Core 1 Router Configuration on page 1133
- Core 2 Router Configuration on page 1133

Core 1 Router Configuration

The following example shows the *EAPS* and VPLS configuration for the Core 1 router.

```
//create the EAPS protected VLAN and the VPLS service vlan
create vlan v1
configure v1 add ports 2:3  // port 2:3 is the port connected to dist 1
create eaps eaps2
configure eaps eaps2 mode transit
configure eaps eaps2 primary port 2:1  // port going to core 2
configure eaps eaps2 secondary port 2:3
configure eaps eaps2 add control vlan ctrl2   // eaps control vlan
configure eaps eaps2 add protected vlan v1
create eaps shared-port 2:1
configure eaps shared-port 2:1 mode controller
configure eaps shared-port 2:1 link-id 888
enable eaps2
create vpls vpls1 fec-id-type pseudo-wire 2
configure vpls vpls1 add service vlan v1
configure vpls vpls1 redundancy eaps
configure vpls vpls1 add peer <core2>
configure vpls vpls1 add peer <core3>
configure vpls vpls1 add peer <core4>
```

Core 2 Router Configuration

The following example shows the *EAPS* and VPLS configuration for the Core 2 router.

```
//create the EAPS protected VLAN and the VPLS service vlan
create vlan v1
configure v1 add ports 2:9  // port 2:9 is the port connected to dist 5
create eaps eaps2
configure eaps eaps2 mode transit
configure eaps eaps2 primary port 2:1 // port going to core 1
configure eaps eaps2 secondary port 2:9
configure eaps eaps2 add control vlan ctrl2
configure eaps eaps2 add protected vlan v1
create eaps shared-port 2:1
configure eaps shared-port 2:1 mode partner
configure eaps shared-port 2:1 link-id 888
enable eaps2
create vpls vpls1 fec-id-type pseudo-wire 2
configure vpls vpls1 add service vlan v1
configure vpls vpls1 redundancy eaps
```

```
configure vpls vpls1 add peer <core1>
configure vpls vpls1 add peer <core3>
configure vpls vpls1 add peer <core4>
```

Configuring H-VPLS

H-VPLS is configured at the edge of the network. The core of the network supports H-VPLS and is configured as described in Configuring MPLS Layer-2 VPNs (VPLS and VPWS) on page 1126. To configure H-VPLS, you need to configure the H-VPLS spoke nodes and the PE core nodes that peer with the spoke nodes.

Configuring H-VPLS Spoke Nodes

- To configure an MTU as an H-VPLS spoke node, use the following command:

  `configure l2vpn [`**`vpls`** *`vpls_name`* `|` **`vpws`** *`vpws_name`*`] `**`add peer`** *`ipaddress`* `{{`**`core`** `{`**`full-mesh`** `|` **`primary`** `|` **`secondary`**`} |` **`spoke`**`}`

  Use the core primary and core secondary command options as needed. The core primary option specifies that the spoke node peer is a core node and that the link between the peers is the primary spoke. The core secondary option specifies that the spoke node peer is a core node and that the link between the peers is the secondary spoke.
- To delete an H-VPLS spoke node, use the following command:

  `configure l2vpn [`**`vpls`** *`vpls_name`* `|` **`vpws`** *`vpws_name`*`] `**`delete peer`** `[`*`ipaddress`* `|` **`all`**`]`

Configuring H-VPLS Core Nodes

- To configure a VPLS core node as an H-VPLS core node, use the following command:

  `configure l2vpn [`**`vpls`** *`vpls_name`* `|` **`vpws`** *`vpws_name`*`] `**`add peer`** *`ipaddress`* `{{`**`core`** `{`**`full-mesh`** `|` **`primary`** `|` **`secondary`**`} |` **`spoke`**`}`

  Use the spoke command option to specify that the peer is an H-VPLS spoke node. When the H-VPLS spoke and core peers are configured, VPLS communications can be established between them.

  Note
  To enable communications from the H-VPLS spoke across the VPLS network, the H-VPLS core node must also be configured to peer with the other VPLS nodes.

- To delete an H-VPLS core node, use the following command:

  `configure l2vpn [`**`vpls`** *`vpls_name`* `|` **`vpws`** *`vpws_name`*`] `**`delete peer`** `[`*`ipaddress`* `|` **`all`**`]`

Configuring the MAC Address Withdrawal Feature

The MAC address withdrawal feature is enabled by default.

- To disable this feature, use the following command:
  `disable l2vpn vpls peer [ipaddress | all] fdb send-mac-withdrawal`
- To enable this feature after it has been disabled, use the following command:
  `enable l2vpn vpls peer [ipaddress | all] fdb send-mac-withdrawal`

Displaying H-VPLS Configuration Information

- To display H-VPLS configuration information, use the following command:
  `show [ {l2vpn} vpls {vpls_name} | l2vpn vpws {vpws_name} | l2vpn ] {peer ipaddress} {detail} | summary }`

Configuring Protected VPLS

- To configure a protected VPLS, use the following command:
  `create esrp esrp_domain {type [vpls-redundancy | standard]}`
  `configure {l2vpn} vpls vpls_name redundancy [esrp esrpDomain | eaps | stp]`
- To unconfigure a protected VPLS, use the following command:
  `unconfigure {l2vpn} vpls vpls_name redundancy [eaps | esrp | stp]`
- To display information about the protected VPLS, use the following command:
  `show esrp { {name} | {type [vpls-redundancy | standard]} }`

Configuring RSVP-TE

Enabling and Disabling RSVP-TE on the Switch

- To enable RSVP-TE on a switch, use the following command:
  `enable mpls protocol rsvp-te`
- To disable RSVP-TE on a switch, use the following command:
  `disable mpls protocol rsvp-te`

Note
MPLS must be globally enabled before RSVP-TE can become operational. For more information, see Enabling and Disabling MPLS on an LSR on page 1108.

Enabling and Disabling RSVP-TE on a VLAN

- To enable RSVP-TE on one or all *VLAN*s, use the following command:
  `enable mpls rsvp-te [{vlan} vlan_name | vlan all]`

- To disable RSVP-TE on one or all VLANs, use the following command:

  ```
  disable mpls rsvp-te te [{vlan} vlan_name | vlan all]
  ```

  Disabling RSVP-TE on a VLAN causes all TE LSPs using that interface to be released, and prevents TE LSPs from being established or accepted on the specified VLAN.

Configuring RSVP-TE Protocol Parameters

- To configure RSVP-TE protocol parameters, use the following command:

  ```
  configure mpls rsvp-te timers session[{bundle-message-
  time bundle_message_milliseconds} {hello-keep-multiplier
  hello_keep_number} {hello-time hello_interval_seconds}{refresh-keep-
  multiplier refresh_keep_number} {refresh-time refresh_seconds}
  {summary-refresh-time summary_refresh_milliseconds}] [{vlan} vlan_name
  | vlan all]
  ```

  This command configures the RSVP-TE protocol parameters for the specified *VLAN*. The RSVP-TE keyword all indicates that the configuration changes apply to all RSVP-TE enabled VLANs.

  The hello-interval time specifies the RSVP hello packet transmission interval. The RSVP hello packet is used by the switch to detect when an RSVP-TE peer is no longer reachable. If an RSVP hello packet is not received from a peer within [hello-interval * keep-multiplier] seconds, the peer is declared down and all RSVP sessions to and from that peer are torn down. The default hello-interval time is zero, indicating that RSVP hellos are not enabled. The hello-interval can be set to any value between zero and 60 seconds.

  The refresh-time parameter specifies the interval for sending refresh path messages. RSVP refresh messages provide soft state link-level keep-alive information for previously established paths and enable the switch to detect when an LSP is no longer active. RSVP sessions are torn down if an RSVP refresh message is not received from a neighbor within [(keep-multiplier + 0.5) * 1.5 * refresh-time] seconds. The default refresh-time is 30 seconds and the default keep-multiplier value is three. The minimum and maximum refresh-time values are one and 36,000 seconds (or ten hours), respectively. The minimum and maximum keep-multiplier values are one and 255, respectively.

  The bundle-time, specified in tenths of a second, indicates the maximum amount of time a transmit buffer is held so that multiple RSVP messages can be bundled into a single PDU. The default bundle-time is zero, indicating that RSVP message bundling is not enabled. The bundle-time value can be set to any value between zero and 30 (or 3 seconds).

  The summary-refresh-time, specified in tenths of a second, indicates the time interval for sending summary refresh RSVP messages. The summary-refresh-time must be less than the configured refresh-time. The default summary-refresh-time is zero, indicating that no summary refresh RSVP messages are sent. The summary-refresh-time value may be set to any value between zero to 100 (or 10 seconds).

If configured, the bundled and summary refresh RSVP messages are only sent to RSVP-TE peers supporting RSVP refresh reduction.

Creating or Deleting an RSVP-TE LSP

- To create an RSVP-TE LSP, use the following command:

  ```
  create mpls rsvp-te lsp lsp_name destination ipaddress
  ```

  The *lsp_name* parameter is a character string that identifies the LSP within the switch. The *lsp_name* string must begin with an alphabetic character and can contain up to 31 additional alphanumeric characters. The LSP is not signaled until at least one path is added. See Adding a Path to an RSVP-TE LSP on page 1141.
- To delete an RSVP-TE LSP, use the following command:

  ```
  delete mpls rsvp-te lsp [lsp_name | all]
  ```

  Deleting an LSP name disassociates all configured paths with this LSP and all configuration information for the LSP name is deleted. If the LSP has been specified for use by a static route or a VPLS, that configuration information is also deleted. If you specify the all keyword, all LSPs are deleted.

Creating an RSVP-TE Path

- To create an RSVP-TE routed path resource, use the following command:

  ```
  create mpls rsvp-te path path_name
  ```

  The *path_name* parameter is a character string that is to used to identify the path within the switch. The *path_name* string must begin with an alphabetic character, and can contain up to 31 additional alphanumeric characters. The RSVP-TE LSP is not signaled along the path until an LSP adds the specified path name. The maximum number of configurable paths is 1000.
- To delete an RSVP-TE path, use the following command:

  ```
  delete mpls rsvp-te path [path_name | all]
  ```

  This command deletes a configured *MPLS* RSVP-TE routed path with the specified *path_name*. All associated configuration information for *path_name* is deleted. A path cannot be deleted as long as the *path_name* is associated with an LSP. If the all keyword is specified, all paths not associated with an LSP are deleted.

Configuring an Explicit Route

The routed path for an RSVP-TE LSP can be described by a configured sequence of the LSRs and/or subnets traversed by the path. Each defined LSR or subnet represents an ERO subobject. Up to 64 subobjects can be added to each path name. LSRs and/or subnets can be either included or excluded.

- To add an RSVP-TE explicit route, use the following commands:
  - To specify an LSR or subnet to be included in the path calculation:

```
configure mpls rsvp-te path path_name add ero {include} ipNetmask
[strict|loose] {order number}
```

- To specify an LSR or subnet to be excluded from the path calculation:

```
configure mpls rsvp-te path path_name add ero exclude ipNetmask
{order number}
```

These commands add an IP address to the explicit route object (ERO) for the specified path name. The include keyword is optional and the default behavior is to define an ERO that must be included in the path. The exclude keyword allows a path to be created that must avoid certain subnets. This can be useful when defining redundant LSPs or paths that must avoid the path of other LSPs or paths.

The ipaddress keyword identifies an LSR using either a /32 address, which may represent an LSR router ID, loopback address, or direct router interface, or an IP prefix, which represents a directly connected subnet. Each IP address or prefix is included in the ERO as an IPv4 subobject.

For EROs that are configured to be included in the path calculation, if the IP address is specified as strict, the strict subobject must be topologically4 adjacent to the previous subobject as listed in the ERO. If the IP address is specified as loose, the loose subobject is not required to be topologically adjacent to the previous subobject as listed in the ERO. If omitted, the default subobject attribute is loose.

For EROs that are configured to be excluded in the path calculation, a given subnet is avoided if any address on that subnet is specified.

If the subobject matches a direct router interface or a directly attached subnet, the switch verifies that the path message is received on the matching router interface. The LSR path order is optionally specified using the order keyword. The order number parameter is an integer value from 1 to 65535. IP prefixes with a lower order number are sequenced before IP prefixes with a higher number. You can specify multiple addresses and assign them an order number. The order number determines the path that the LSP follows. Thus, the LSP follows the configured path of the IP prefix with the order value from low to high. If the order keyword is not specified, the number value for the LSR defaults to a value 100 higher than the current highest number value. For excluded nodes, the order is not important. Any excluded ERO will be always be avoided no matter where it is in the list of ERO objects.

If the list of IP prefixes added to the path does not reflect an actual path through the network topology, the path message is returned with an error from a downstream LSR and the LSP is not established.

The order of a configured subobject can not be changed. The ERO subobject must be deleted and re-added using a different order. If a subobject is added to or deleted from the ERO while the associated LSP is established, the path is torn down and is resignaled using the new ERO. Duplicate ERO subobjects are not allowed. Defining an ERO for the path is optional. If you do not configure an ERO, the path is signaled along the best CSPF calculated path and the ERO is not included in the path message. When the last subobject in the ERO of the path message is reached and the egress IP node of the path has not been reached, the remaining path to the egress node is signaled along

the best CSPF calculated path. Specification of an ERO could lead to undesirable routed paths, so care should be taken when terminating the ERO routed-path definition prior to the configured path egress node.

To delete an RSVP-TE explicit route, use the following command:

```
configure mpls rsvp-te path path_name delete ero [all | ipNetmask |
order number]
```

This command deletes an LSR or subnet from the ERO for the specified path name. The LSR is specified using the *ipaddress*, or *order* parameter. If an LSR is deleted from an ERO while the associated LSP is established, the path is torn down and is resignaled using a new ERO. Use the all keyword to delete the entire ERO from the path name. When there is no configured ERO, the path is no longer required to take an explicit routed path. The path is then signaled along the best CSPF calculated path and no ERO is included in the path message.

Reserving Bandwidth for MPLS

In order to manage the bandwidth used by *MPLS* RSVP-TE LSPs, a configured amount of bandwidth must be reserved for MPLS. Bandwidth in the transmit direction must be reserved on all MPLS interfaces which will be used as egress interfaces for RSVP-TE LSPs requesting bandwidth. Bandwidth in the receive direction can be optionally reserved on all MPLS interfaces that will be used as ingress interfaces for RSVP-TE LSPs requesting bandwidth. Note that the receive bandwidth reservation can be exceeded, and is provided only for tracking and informational purposes.

- The following command is used to reserve bandwidth for MPLS:

  ```
  configure mpls rsvp-te bandwidth committed-rate committed_bps [Kbps |
  Mbps | Gbps] [{vlan} vlan_name | vlan all] {receive | transmit | both}
  ```

  The committed_rate_unit must be specified in Kbps, Mbps, or Gbps. Choosing the **vlan all** option reserves the specified amount of bandwidth on all MPLS interfaces.

  The default reserved value is zero. Therefore, no LSPs requesting bandwidth can be established until the bandwidth has been configured.

Creating and Deleting an RSVP-TE Profile

- To create a traffic engineered LSP profile, use the following command:

  ```
  create mpls rsvp-te profile profile_name {standard}
  ```

  This command creates a configured RSVP-TE profile with the specified profile name. The default profile cannot be deleted. If a profile is associated with a configured LSP, the profile cannot be deleted.
- To delete a traffic engineered LSP profile, use the following command:

  ```
  delete mpls rsvp-te profile [profile_name | all]
  ```

Configuring an RSVP-TE Profile

- To configure an RSVP-TE profile, use the following command:

  ```
  configure mpls rsvp-te profile profile_name {bandwidth [best-effort |
  [{committed-rate committed_bps [Kbps | Mbps | Gbps]} {max-burst-size
  burst_size [Kb | Mb]} {peak-rate peak_bps [Kbps | Mbps | Gbps]}]}
  {hold-priority hold_priority} {mtu [number | use-local-interface]}
  {path-computation [full | partial]} {record [enabled {route-only} |
  disabled]} {setup-priority setup_priority}
  ```

  A profile is a set of attributes that are applied to the LSP when the LSP is configured using the `create mpls rsvp-te lsp` command. A default profile is provided which cannot be deleted, but can be applied to any configured LSP. The profile name for the default profile is default. The default profile parameter values are initially set to their respective default values. The maximum number of configurable profiles is 1000 (one of which is reserved for the default profile).

  The bandwidth parameter specifies the desired reserved bandwidth for the LSP. Any positive integer value is valid. You must append the characters k for kilobits, m for megabits, or g for gigabits, to the bps value to specify the unit of measure. The default bandwidth bps value is zero, which indicates that the *QoS* for the LSP is best effort.

  The max-burst-size and peak-rate parameters are signaled in the sender Tspec object and add further definition of the expected traffic. The mtu parameter is also signaled in the sender Tspec object and defines the expected maximum packet size that is sent over the LSP.

  The setup-priority and hold-priority are optional parameters indicating the LSP priority. During path set up, if the requested bandwidth cannot be reserved through the LSR, the setup-priority parameter is compared to the hold-priority of existing LSPs to determine if any of the existing LSPs need to be preempted to allow a higher priority LSP to be established. Lower numerical values represent higher priorities. The setup-priority range is 0 to 7 and the default value is 7. The hold-priority range is also 0 to 7 and the default value is 0.

  The path-computation keyword specifies computation strategy for calculating a path to the LSP destination. "full" requires the ingress node to full calculate a path to the LSP destination. "partial" allows the ingress node to calculate only part of the path to the LSP destination.

  Note
  Inter area RSVP fast reroute (FRR) is not supported.

  The record keyword is used to enable hop-by-hop path recording. The enabled keyword causes the record route object (RRO) to be inserted into the path message. The RRO is returned in the reserve message and contains a list of IPv4 subobjects that describe the RSVP-TE path. Path recording by default is disabled. When disabled, no RRO is inserted into the path message.

- To delete an RSVP-TE path profile, use the following command:

  ```
  delete mpls rsvp-te profile [profile_name | all]
  ```

This command deletes a configured RSVP-TE profile with the specified profile name. The default profile cannot be deleted. If a profile is associated with a configured LSP, the profile cannot be deleted. If you specify the all keyword, all profiles not associated with an LSP are deleted (except for the default profile).

Adding a Path to an RSVP-TE LSP

- To add a path to an RSVP-TE LSP, use the following command:

  `configure mpls rsvp-te lsp` *`lsp_name`* **`add path`** [*`path_name`* | **`any`**] {**`profile`** *`profile_name`*} {**`primary`** {*`frr_profile_name`*} | **`secondary`**}

  This command adds a configured path to the specified RSVP-TE LSP. The LSP name parameter is a character string that is to be used to identify the LSP within the switch and must have been created previously. The LSP is not signaled until a path is added to the LSP. Up to three paths can be defined for the LSP: one primary and two secondary. All paths are signaled, but only one path is used to forward traffic at any one time. The switch chooses the local *MPLS VLAN* interface from which to signal the LSP. To force an LSP to use a specific local MPLS interface, configure the peer's interface IP address as the first ERO in the associated path. The profile name is optional. If omitted, the default profile is applied to the LSP. The path name can be specified or the LSP can be configured to take any path. For a given LSP, only one path can be configured to take any path through the MPLS network.

  The specified path defaults to primary when no primary path has been configured for the LSP and defaults to secondary if the primary path has been previously configured for the LSP.

  Each *`path_name`* added to an *`lsp_name`* must be unique, but a *`path_name`* can be associated with multiple LSP names.

  All configured primary and secondary paths for the *`lsp_name`* must have the same endpoint IP address. For example, three paths can be configured for the *`lsp_name`*, but all paths should represent different topological paths through the network to the same LSP endpoint.

  Adding a secondary *`path_name`* designates a path as a hot-standby redundant path, used in the event that the primary or the other secondary path cannot be established or fails. Provided the *`path_name`* has not already been established, all paths are signaled as soon as they are associated with an *`lsp_name`*. If the primary *`path_name`* fails, is not configured, or cannot be established after the specified LSP retry-timeout, one of the configured secondary paths becomes the active path for *`lsp_name`*. All of the secondary paths have equal preference; the first one available is chosen. If at any time the primary path is reestablished, *`lsp_name`* immediately switches to using the primary path. If a secondary path fails while in use, the remaining configured secondary path can become the active path for *`lsp_name`*.

- To delete a path from an RSVP-TE LSP, use the following command:

  `configure mpls rsvp-te lsp` *`lsp_name`* **`delete path`** [*`path_name`* | **`any`** | **`all`**]

When you issue this command, the LSP associated with the path is immediately torn down. If the deleted path represents the in-use LSP for *lsp_name* and another secondary path is configured, the LSP immediately fails over to an alternate LSP. Because at least one path must be defined for each LSP, the last configured path cannot be deleted from the LSP.

Setting up Fast-Reroute Protection for an LSP

To create a protected LSP, do the following:

1. Create the LSP as described in Creating or Deleting an RSVP-TE LSP on page 1137.
2. Enable the fast-reroute feature on the LSP using the following command: `configure mpls rsvp-te lsp` *lsp_name* **fast-reroute** [**enable** | **disable**]
3. Create a path for the LSP as described in Creating an RSVP-TE Path on page 1137.
4. Define the path route as described in .
5. If you want to use a custom profile instead of the default profile, create a profile for the protected LSP as described in Creating and Deleting an RSVP-TE Profile on page 1139.
6. If you want to configure the custom profile created in the previous step, configure the profile as described in Configuring an RSVP-TE Profile on page 1140.
7. If you want to use a custom fast-reroute profile instead of using the default fast-reroute profile, create the profile using the following command at the ingress LER: `create mpls rsvp-te profile` *profile_name* **fast-reroute**
8. If you want to configure the custom fast-reroute profile created in the previous step, use the following command at the ingress LER:
   `configure mpls rsvp-te profile` *frr_profile_name* {**bandwidth** *bandwidth_rate_bpsbandwidth_rate_unit*} {**detour** {**hop-limit** *hop_limit_value*} {**bandwidth-protection** [**enabled** | **disabled**]} {**node-protection** [**enabled** | **disabled**]}} {**hold-priority** *hold_priority_value*} {**setup-priority** *setup_priority_value*}
9. Add the path to the protected LSP and select the standard and fast-reroute profiles using the following command:
   `configure mpls rsvp-te lsp` *lsp_name* **add path** [*path_name* | **any**] {**profile** *profile_name*} {**primary** {*frr_profile_name*} | **secondary**}

RSVP-TE Configuration Example

RSVP-TE LSPs comprise profiles, paths, and the actual LSP. This section describes how to configure an RSVP-TE LSP.

Configuring RSVP LSPs is a multi-step process with some optional steps, depending on the specific requirements of the LSP. Conceptually, a number of mandatory elements must be configured to create an RSVP-TE LSP. In addition, you can also configure optional elements. In certain configurations, there are also order dependencies.

The profile contains constraints that you might wish to apply to the LSP. These constraints can affect the path selected across the *MPLS* domain in order to meet those constraints. Examples of profile parameters include bandwidth, setup, and hold priority relative to other configured LSPs.

The path can be used to specify the explicit path across the MPLS domain that the LSP should follow. This is done using EROs. An ERO is an object, sent as part of the LSP setup request (path message) that explicitly specifies the part of the path across the MPLS domain the setup request should follow. You can configure both loose and strict EROs in a path.

Certain elements of configuration are order dependent. For example if you specify a profile or path when creating an LSP, those path or profile definitions must already exist. Similarly a path must exist before an ERO is created, as the ERO is added explicitly to the path.

The typical steps used to configure and verify an RSVP-TE LSP are as follows:

1. Create and configure a path (optional).
2. Reserve bandwidth for the LSP (optional).
3. Create and configure a profile (optional).
4. Create an LSP (mandatory).
5. Add a primary/secondary path to the LSP (mandatory).
6. Add a secondary path to the LSP (optional).
7. Verify LSP status (recommended).

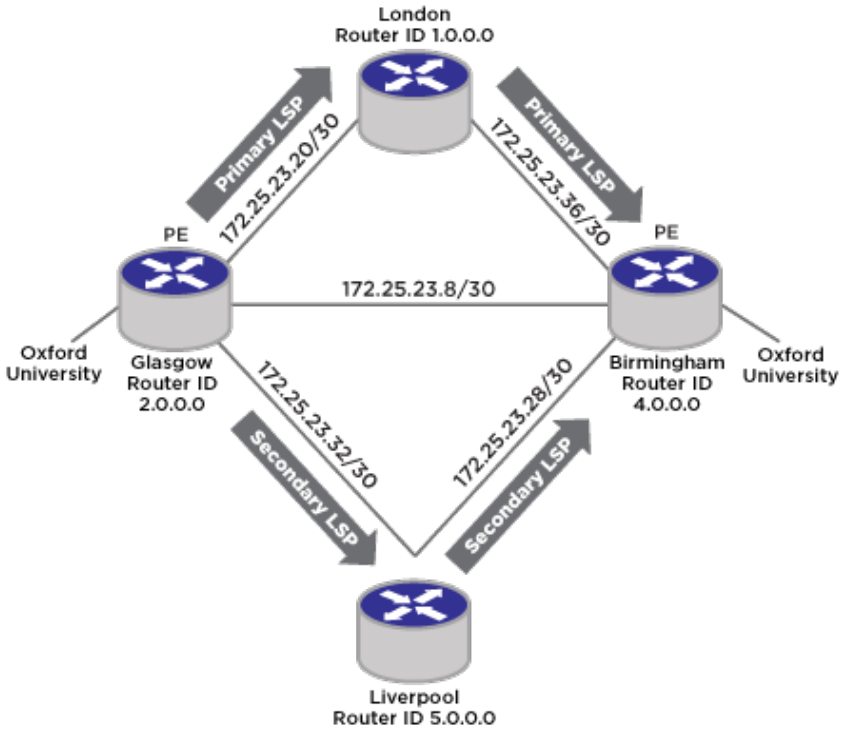


Figure 140: RSVP-TE Configuration Example

The configuration example, shown in Figure 140, creates primary and secondary LSPs between the node Glasgow and the node Birmingham. The steps specifically create an LSP between Glasgow and Birmingham based on an explicitly routed path via London with bandwidth, setup priority, and hold priority profile requirements.

A secondary path is also created which, in the event of failure of a link or node on the primary path, activates the secondary LSP from Glasgow to Liverpool to Birmingham.

Note

Before configuring RSVP-TE LSPs, you need to enable the protocol on the switch, and an initial step of adding RSVP-TE to a *VLAN* must be carried out for all VLANs over which the user wishes RSVP-TE LSPs to be signaled. This is a one-time operation.

A loopback VLAN with the LSR-ID should be added to MPLS to allow RSVP-TE LSPs to be established to the LSR-ID.

The following commands configure RSVP-TE for the switch and add RSVP signaling capabilities to the specified VLANs:

```
enable mpls
enable mpls protocol rsvp-te
configure mpls add vlan loopback
configure mpls add vlan gla-lon
enable mpls rsvp-te vlan gla-lon
enable mpls vlan gla-lon
configure mpls add vlan gla-liv
enable mpls rsvp-te vlan gla-liv
enable mpls vlan gla-liv
```

The following commands reserve bandwidth for RSVP-TE LSPs on these MPLS interfaces:

```
configure mpls rsvp-te bandwidth committed-rate 20 Mbps gla-lon
configure mpls rsvp-te bandwidth committed-rate 20 Mbps gla-liv
```

The following commands create and configure an LSP profile named Glasgow-Birmingham-pro.

LSPs that use the Glasgow-Birmingham-pro profile are signaled with a reserved bandwidth of 10 Mbps and an LSP setup and hold priority of 5.

```
create mpls rsvp-te profile Glasgow-Birmingham-pro
configure mpls rsvp-te profile Glasgow-Birmingham-pro bandwidth committed-rate 10 m
configure mpls rsvp-te profile Glasgow-Birmingham-pro setup-priority 5 hold-priority 5
```

The following commands define the primary and secondary paths between Glasgow and Birmingham:

```
create mpls rsvp-te path Glasgow-Birmingham-pri-path
create mpls rsvp-te path Glasgow-Birmingham-sec-path
```

The following commands pin each path to an LSR, such that each path takes a different route to the endpoint 4.0.0.0.

Path Glasgow-Birmingham-pri-path is routed through LSR 1.0.0.0 and path Glasgow-Birmingham-sec-path is routed through LSR 5.0.0.0.

```
configure mpls rsvp-te path Glasgow-Birmingham-pri-path add ero 1.0.0.0/32 loose
configure mpls rsvp-te path Glasgow-Birmingham-sec-path add ero 5.0.0.0/32 loose
```

The following commands create one RSVP-TE LSP with one primary and one secondary or backup path.

Each path uses the same profile.

```
create mpls rsvp-te lsp Glasgow-Birmingham-lsp destination 4.0.0.0
configure mpls rsvp lsp Glasgow-Birmingham-lsp add path Glasgow-Birmingham-pri-path
profile Glasgow-Birmingham-pro primary
configure mpls rsvp lsp Glasgow-Birmingham-lsp add path Glasgow-Birmingham-sec-path
profile Glasgow-Birmingham-pro secondary
```

Note
The secondary LSP is signaled, however it remains in a standby state unless the primary path becomes unavailable.

By default, a VPLS pseudowire flows over any available LSP.

However, a VPLS pseudowire can be specifically directed to use a configured RSVP-TE based LSP. Configuration is no different from configuring an LDP-based VPLS pseudowire, except that the RSVP-TE LSP is explicitly specified. The following command specifically directs a VPLS pseudowire to use a previously configured RSVP-TE LSP:

```
configure vpls Glasgow-Birmingham-cust1 peer 4.0.0.0 add mpls lsp Glasgow-Birmingham-
lsp
```

Troubleshooting MPLS

The ExtremeXOS software includes multiple mechanisms for detecting and reporting problems.

Many failures generate an *SNMP* trap or log an error message. To find out more about a problem, you can enter show commands and review the flag states in the command output. The software also includes some *MPLS* troubleshooting tools, which are described in the following sections.

Using LSP Ping

To assist with problem determination in an *MPLS* network, LSP ping support is included as a CLI ping option in the ExtremeXOS software. The LSP ping support is based on draft-ietf-mpls-lsp-ping-13.txt. This draft includes support for both connectivity verification and fault isolation for transport LSPs. Connectivity verification is supported using a modified ping packet that is sent over the specified transport LSP.

LSP ping is designed to catch failures where a transport LSP appears to be operational but is actually not functioning correctly. LSP data plane corruption is far less likely to occur than an LSP control plane failure, but the LSP ping is also useful for detecting possible latency issues.

- To send MPLS ping packets over an LSP, enter the following command:
  `ping mpls lsp [lsp_name | any host | prefix ipNetmask] {reply-mode [ip | ip-router-alert]} {continuous | count count} {interval interval} {start-size start-size {end-size end-size}} {ttl ttl} {{from from} {next-hop hopaddress}}`

MPLS pings are sent to the well-known UDP port number 3503 with an IP in the 127.0.0.0/8 IP subnet. The source IP address is set to the sender.

The time stamp field is supported for calculating round trip times and is accurate to 1/100 of a second. When replying to a ping, the LSP ping response (MPLS echo reply) sequence number and time-stamp fields are set to the LSP ping request (MPLS echo request) values. One MPLS echo response is sent for each MPLS echo request received. An MPLS echo reply is sent out-of-band as a natively IP routed IPv4 UDP packet. The normal IP routed path might or might not use an LSP.

To reduce the possibility of fragmentation problems on the return path, MPLS echo reply packets do not include any padding that was sent in the MPLS echo request. Because each LSP is unidirectional, the return path is not directly relevant for verification of the LSP's functionality. What is important is that the LSP ping results are returned to the source of the MPLS echo request.

Using LSP Trace

Transport LSP fault isolation is supported using the LSP trace feature. When the control plane detects a transport LSP failure, the LSR can switch to another LSP if one is available. When the failure is detected by LSP ping, you can use LSP trace to try to isolate the fault.

- To start an LSP trace, use the following command:

  ```
  traceroute mpls lsp [lsp_name | any host | prefix ipNetmask]
  {reply-mode [ip | ip-router-alert]} {{from from} {ttl ttl} {next-hop
  hopaddress}}
  ```

Using the Health Check VCCV Feature

Health check Virtual Circuit Connectivity Verification (VCCV) can be used as a network diagnostic tool or as a network fault-alert tool, and can be configured on up to 16 VPLS domains. Connectivity between VPLS peers can be verified using the health check VCCV feature, which is defined in RFC 5085, Pseudowire Virtual Circuit Connectivity Verification (VCCV).

This implementation uses the following components defined in that RFC:

- VCCV Control Channel Type: *MPLS* Router Alert Label
- VCCV Control Verification Type: LSP Ping

Health check uses the LSP ping capability to verify connectivity. When the PW is set up, the two peers negotiate the VCCV capabilities, and if they establish a common set, health check becomes operational. If the VCCV capabilities do not match, health check cannot operate.

Health check operates in a single direction, so health checking should be enabled on the LSRs at both ends of the pseudowire in order to verify that traffic can flow bi-directionally between two VPLS peers. For multi-peer full-mesh or hierarchical VPLS networks, VCCV should be enabled on all VPLS peers to verify the entire VPLS network.

VCCV sends health check packets at regular intervals (the default interval is 5 seconds). If health check reaches the threshold for missed responses (the default fault-multiplier is 4), health check logs a message in EMS at the Warning level. Note that this log is not seen with the default log settings and no *SNMP* traps are sent. A health check failure does not change the state of the PW, which could remain operationally up and continue to support traffic, depending on the actual problem.

- To enable health check, use the following commands:

  ```
  enable l2vpn [vpls vpls_name | vpws vpws_name] health-check vccv
  ```

- To configure health check, use the following command:

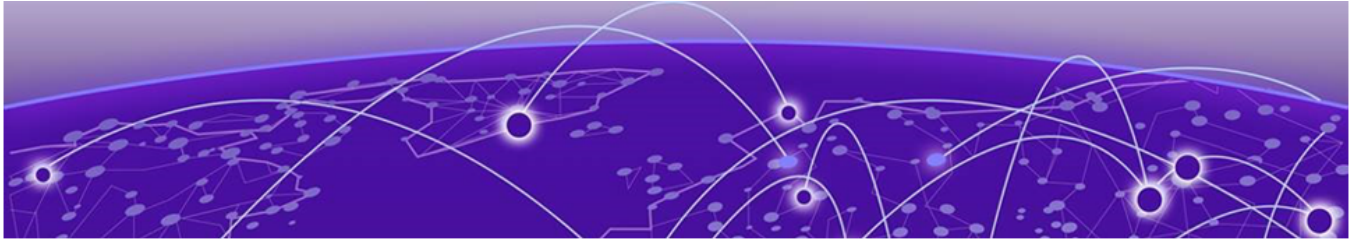
  ```
  configure l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]] health-check vccv {interval interval_seconds} {fault-multiplier fault_multiplier_number}
  ```

- View VPLS configuration information, including the health check feature configuration.

  ```
  show [ {l2vpn} vpls {vpls_name} | l2vpn vpws {vpws_name} | l2vpn ] {peer ipaddress} {detail} | summary }
  ```

- To disable health check, use the following command:

  ```
  disable l2vpn [vpls [vpls_name | all] | vpws [vpws_name | all]] health-check vccv
  ```



MSDP

MSDP Overview on page 1148
PIM Border Configuration on page 1149
MSDP Peers on page 1149
MSDP Mesh-Groups on page 1152
Anycast RP on page 1153
SA Cache on page 1155
SNMP MIBs on page 1156

This chapter introduces MSDP (Multicast Source Discovery Protocol), an interdomain multicast protocol used to connect multiple multicast routing domains that run PIM-SM (Protocol Independent Multicast-Sparse Mode). This chapter discusses the features and configuration for PIM border, MSDP peers, mesh groups, anycast RP, SA cache, redundancy, and *SNMP (Simple Network Management Protocol)* MIBs.

Note
For more information about MSDP, refer to RFC 3618.

MSDP Overview

MSDP (Multicast Source Discovery Protocol) is an interdomain multicast protocol used to connect multiple multicast routing domains that run Protocol Independent Multicast-Sparse Mode (PIM-SM).

MSDP speakers are configured on each PIM-SM domain. These speakers establish a peering relationship with other MSDP speakers through secured TCP connections. When the source sends traffic, the MSDP speaker learns about the source through its Rendezvous Point (RP). In turn, the RP advertises the source to its peers through Source Active (SA) messages. The peers receive these advertisements and inform their RPs about the presence of the active source in the other PIM-SM domain, which triggers the normal PIM operation in the corresponding domains.

For example, as businesses expand and networks grow in size, it might become necessary to connect PIM domains to allow multicast applications to reach other offices across the network. MSDP simplifies this process by providing a mechanism to connect those multicast routing domains without reconfiguring existing domains. Each PIM domain remains separate and has its own RP. The RP in each domain establishes an MSDP peering relationship over a TCP connection either with RPs in other domains or with border routers leading to other domains. When an RP learns about a new multicast source in its own domain (using the normal PIM registration process), it then

sends a SA message to all of its MSDP peers, letting them know about the new stream. In this way, the network can receive multicast traffic from all over the network without having to reconfigure each existing PIM domain.

Supported Platforms

MSDP is supported on all platforms running a minimum software version of ExtremeXOS 12.0 with the Premier or Extreme Platform ONE license.

Our implementation of MSDP is compliant with RFC 3618 and RFC 3446, and compatible with other devices that are compliant with these standards.

Limitations

The limitations of are as follows:

- There is no support for MSDP operating with SA cache disabled (transit node). MSDP will always cache/store received SA messages.
- There is no support for logical RP.
- There is no support for MSDP on user-created virtual routers (VRs).
- routes are not used for peer-RPF checking. So, our implementation of MSDP does not exactly conform to rule (iii) in section 10.1.3 of RFC 3618. However, our implementation of MSDP uses / for peer-RPF checking as per rule (iii) in section 10.1.3.
- Read-write/read-create access is not supported on MSDP MIB objects.

PIM Border Configuration

To create a PIM-SM domain for *MSDP*, you must restrict the reach of Bootstrap Router (BSR) advertisements by defining a *VLAN (Virtual LAN)* border. BSR advertisements are not sent out of a PIM interface configured as a VLAN border, thereby defining a PIM domain for MSDP.

To configure a PIM VLAN border, use the command:

```
configure pim vlan_name border
```

MSDP Peers

MSDP peers exchange messages to advertise active multicast sources. The peer with the higher IP address passively listens to a well-known port number and waits for the side with the lower IP address to establish a Transmission Control Protocol (TCP) connection on port 639. When a PIM-SM RP that is running MSDP becomes aware of a new local source, it sends an SA message over the TCP connection to its MSDP peer. When the SA message is received, a peer-RPF check is performed to make sure the peer is toward the originating RP. If so, the RPF peer floods the message further. If not, the SA message is dropped and the message is rejected.

- Configure an MSDP peer using the command:

  ```
  create msdp peer remoteaddr {remote-as remote-AS} {vr vrname}
  ```

- Delete an MSDP peer using the command:
  `delete msdp peer [all | remoteaddr] {vr vr_name}`
- Display configuration and run-time parameters about an MSDP peer using the command:
  `show msdp [peer {detail} | {peer} remoteaddr] {vr vr_name}`

MSDP Default Peers

You can configure a default peer to accept all SA messages. Configuring a default peer simplifies the peer-RPF checking of SA messages. If no policy is specified, the current peer is the default RPF peer for all SA messages.

When configuring a default peer, you can also specify an optional policy filter. If the peer-RPF check fails, and a policy filter is configured, the default peer rule is applied to see if the SA message should be accepted or rejected.

You can configure multiple default peers with different policies. However, all default peers must either be configured with a default policy or not. A mix of default peers, with a policy and without a policy, is not allowed.

- Configure an *MSDP* default peer, and optional policy filter using the command:
  `configure msdp peer [remoteaddr | all] default-peer {default-peer-policy filter-name} {vr vrname}`
- Remove the default peer using the command:
  `configure msdp peer [remoteaddr | all] no-default-peer {vr vrname}`
- Verify that a default peer is configured using the command:
  `show msdp [peer {detail} | {peer} remoteaddr] {vr vr_name}`

Peer Authentication

MSDP supports TCP RSA Data Security, Inc. *MD5 (Message-Digest algorithm 5)* Message-Digest Algorithm authentication (RFC 2385) to secure control messages between MSDP peers. You must configure a secret password for an MSDP peer session to enable TCP RSA Data Security, Inc. MD5 Message-Digest Algorithm authentication. When a password is configured, MSDP receives only authenticated MSDP messages from its peers. All MSDP messages that fail TCP RSA Data Security, Inc. MD5 Message-Digest Algorithm authentication are dropped.

- Configure TCP RSA Data Security, Inc. MD5 Message-Digest Algorithm authentication on an MSDP peer using the command:
  `configure msdp peer [remoteaddr | all] password [none | {encrypted} tcpPassword] {vr vrname}`
- Remove the password using the command:
  `configure msdp peer {all | remoteaddr} password none`

  `The password displays in encrypted format and cannot be seen as simple text. Additionally, the password is saved in encrypted format.`
- Display the password in encrypted format using the command:
  `show msdp [peer {detail} | {peer} remoteaddr] {vr vr_name}`

Policy Filters

You can configure a policy filter to control the flow of SA messages going to or coming from an *MSDP* peer. For example, policy filters can help mitigate state explosion during denial of service (DoS) or other attacks by limiting what is propagated to other domains using MSDP.

You can use the following policy attributes in an SA filter policy. All other attributes are ignored:

- Match:
  - multicast-group
  - multicast-source
  - pim-rp
- Set:
  - permit
  - deny

The following shows some examples of policy files taking into account multicast-source with permit/deny action:

```
entry policy2 {
     if match any{
         nlri 10.10.10.0/24;
      }
      then {
          deny;
      }
}

entry policy1 {
     if match any{
         nlri 10.10.10.0/24;
         nlri 20.20.20.0/24 ;
      }                        <= permit these sources
      then {
          permit;
      }
}
```

- To configure an incoming or outgoing policy filter for SA messages:

  `configure msdp peer [`*`remoteaddr`* `|` **`all`**`] ` **`sa-filter`** `[`**`in`** `|` **`out`**`] [`*`filter-name`* `|` **`none`**`] {`**`vr`** *`vr_name`*`}`

- To remove a policy filter for SA messages, use the none keyword:

  `configure msdp [{`**`peer`**`} `*`remoteaddr`* `|` **`peer all`**`] ` **`sa-filter`** `[`**`in`** `|` **`out`**`] ` **`none`**

- To verify that a policy filter is configured on an MSDP peer:

  `show msdp [`**`peer`** `{`**`detail`**`} | {`**`peer`**`} `*`remoteaddr`*`] {`**`vr`** *`vr_name`*`}`

SA Request Processing

You can configure the router to accept or reject SA request messages from a specified *MSDP* peer or all peers. If an SA request filter is specified, only SA request messages from those groups permitted are accepted. All others are ignored.

- Configure the router to accept SA request messages from a specified MSDP peer or all peers using the command:

  `enable msdp [{peer} remoteaddr | peer all] process-sa-request {sa-request-filter filter-name } {vr vr_name}`

- Configure the router to reject SA request messages from a specified MSDP peer or all peers using the command:

  `disable msdp [{peer} remoteaddr | peer all] process-sa-request {vr vrname}`

- Display configuration and run-time parameters about MSDP peers using the command:

  `show msdp [peer {detail} | {peer} remoteaddr] {vr vr_name}`

MSDP Mesh-Groups

MSDP can operate in a mesh-group topology. A mesh-group limits the flooding of SA messages to neighboring peers. In a mesh-group, every MSDP peer must be connected to every other peer in the group. In this fully-meshed topology, when an SA message is received from a member of the mesh-group, the SA message is always accepted, but not flooded to other members of the same group. Because MSDP peers are connected to every other peer in the mesh-group, an MSDP peer is not required to forward SA messages to other members of the same mesh-group. However, SA messages are flooded to members of other mesh-groups. An MSDP mesh-group is an easy way to implement inter-domain multicast, as it relaxes the requirement to validate looping of MSDP control traffic (that is, peer-RPF checking is not required). Consequently, SA messages do not loop in the network.

Note
We recommend that you configure anycast RP peers in a mesh topology.

- Configure an MSDP mesh-group using the command:

  `create msdp mesh-group mesh-group-name {vr vrname}`

- Remove an MSDP mesh-group using the command:

  `delete msdp mesh-group mesh-group-name {vr vrname}`

- Display information about an MSDP mesh-group using the command:

  `show msdp [mesh-group {detail} | {mesh-group} mesh-group-name] {vr vrname}`

- Configure a peer to be a member of an MSDP mesh-group using the command:

  `configure msdp peer [remoteaddr | all] mesh-group [mesh-group-name | none] {vr vrname}`

- Remove a peer from an MSDP mesh-group using the command:

  `configure msdp [{peer} remoteaddr | peer all] mesh-group none {vr vrname}`

Anycast RP

Anycast RP is an application of *MSDP* that allows multiple RPs to operate simultaneously in a PIM-SM domain. Without anycast RP, multiple routers can be configured as candidate RPs, but at any point in time, only one router can serve as RP. Because anycast RP allows multiple RPs to be simultaneously active, anycast RP provides both load sharing and redundancy, as each RP serves the receivers that are closest to it in the network and can take over for additional receivers if another RP fails.

Note
Starting with ExtremeXOS 31.1, you can use Anycast RP without MSDP using the Anycast RP Using PIM (*RFC 4610*) feature.

In an anycast RP topology, all RPs in a PIM-SM domain are configured with the same IP address on a loopback *VLAN*. The loopback VLAN IP address should have a 32-bit mask, so that it specifies a host address. All the routers within the PIM-SM domain select the nearest RP for each source and receiver. If the senders and receivers within the PIM-SM domain are distributed evenly, the number of senders that register with each RP is approximately equal.

Another requirement of the anycast RP topology is that MSDP must run on all RPs in the PIM-SM domain, so all RPs are also MSDP peers. We recommend that you configure an MSDP mesh connection between all MSDP peers in the domain.

Whenever any multicast source becomes active, this information is sent in an MSDP SA message to the other MSDP peers in the domain, announcing the presence of the source. If any RP within the domain fails, the IP routing protocol mechanism ensures that next available RP is chosen. If a sender registers with one RP and a receiver joins another RP, the information shared through MSDP enables PIM-SM to establish an SPT between the receiver and the source.

Note
We recommend that you configure anycast RP peers in a mesh topology.

The exchange of information in an anycast RP process works as follows:

- When the first-hop router sends a PIM Register message to the nearest RP, the PIM router checks to see if the nearest RP is the RP for the group.
- If the nearest RP is the RP for the group, an MSDP SA message is created and forwarded to the other MSDP peers.
- The MSDP SA message includes the configured originator ID, which is a mandatory configuration component.
- Each remote peer checks the RPF of the originator ID address and informs the PIM process on that remote router about active multicast sources.
- Remote receivers get data packets through the remote shared tree, and can then switch over to the SPT by sending join messages directly towards the source.

To configure anycast RP, do the following at each anycast RP router:

1. Create and configure a loopback VLAN using the commands:

   ```
   create vlan vlan_name {tag tag } {description vlan-description } {vr
   name }

   enable loopback-mode vlan vlan_name
   ```

2. Assign the anycast RP address to the loopback VLAN with a 32 bit subnet mask using the command:

   ```
   configure {vlan} vlan_name ipaddress [ipaddress {ipNetmask} | ipv6-
   link-local | {eui64} ipv6_address_mask]
   ```

   Note
   The anycast RP address must be unique to the loopback VLAN and be the same on all anycast RP peers. It must not match the router IP address, the PIM BSR address, or any other IP addresses used by the router or any other network devices.

3. Enable IP forwarding and IP multicast forwarding on the loopback VLAN using the commands:

   ```
   enable ipforwarding {ipv4 | broadcast} {vlan vlan_name}

   enable ipmcforwarding {vlan name}
   ```

4. Add the loopback VLAN into the unicast routing domain using the appropriate command for your unicast routing protocol:

   ```
   configure ospf add vlan vlan-name area area-identifier link-type [auto
   | broadcast | point-to-point] {passive}

   configure rip add vlan [vlan_name | all]

   configure isis add [vlan all | {vlan} vlan_name] area area_name {ipv4
   | ipv6}
   ```

5. Add the loopback VLAN into the PIM-SM domain and configure it as an RP using the commands:

   ```
   configure pim {ipv4 | ipv6} add vlan [vlan-name | all] {dense |
   sparse} {passive}

   configure pim {ipv4 | ipv6} crp static ip_address [none | policy]
   {priority [0-254]}
   ```

6. Enable MSDP and establish a peering relationship with similar anycast RP neighbors using the commands:

   ```
   create msdp peer remoteaddr {remote-as remote-AS} {vr vrname}

   configure msdp peer [remoteaddr | all] password [none | {encrypted} tcpPassword] {vr vrname}

   configure msdp peer remoteaddr description {peer-description} {vr vrname}

   enable msdp [{peer} remoteaddr | peer all] {vr vr_name}

   enable msdp {vr vrname}
   ```

7. Configure a unique originator ID for each anycast RP peer using the command:

   ```
   configure msdp originator-id ip-address {vr vrname}
   ```

SA Cache

As an *MSDP* router learns of new sources either through a PIM-SM Source-Register (SR) message or SA message from its RPF peer, it creates an entry in SA cache (or refreshes the entry if it is already there) and forwards this information to its peers. These entries are refreshed by periodic SA messages received from the MSDP peers. If these entries are not refreshed within six minutes, they will time out. When a PIM-SM RP detects that the source is no longer available it informs MSDP, which in turn removes the SA information from the local database.

Caching makes it easy for local receivers to know immediately about inter-domain multicast sources and to initiate building a source tree towards the source. However, maintaining a cache is heavy both in CPU processing and memory requirements.

Note
Our implementation of MSDP does not support operating with local cache disabled.

- Remove an SA cache server.

  ```
  unconfigure msdp sa-cache-server {vr vrname}
  ```

  As MSDP uses the flood-and-join model to propagate information about sources, there is a restriction that no more than two advertisements per cache entry will be forwarded per advertisement interval. This is helpful in reducing an SA message storm and unnecessarily forwarding them to peers.

  By default, the router does not send SA request messages to its MSDP peers when a new member joins a group and wants to receive multicast traffic. The new member simply waits to receive SA messages, which eventually arrive.

- Configure the MSDP router to send SA request messages immediately to the MSDP peer when a new member becomes active in a group.

  ```
  configure msdp sa-cache-server remoteaddr {vr vr_name}
  ```

- Purge all SA cache entries.

  ```
  clear msdp sa-cache {{peer} remoteaddr | peer all} {group-address grp-addr} {vr vrname}
  ```

- Display the SA cache database.

  ```
  show msdp [sa-cache | rejected-sa-cache] {group-address grp-addr} {source-address src-addr} {as-number as-num} {originator-rp originator-rp-addr} {local} {peer remoteaddr} {vr vrname}
  ```

Maximum SA Cache Entry Limit

You can configure a limit on the maximum number of SA cache entries that can be stored in the cache database. Once the number of SA cache entries exceeds the pre-configured limit, any newly received cache entries are discarded. You can configure the limit on a per-peer basis. By default, no SA message limit is set. The router can receive an unlimited number of SA entries from an *MSDP* peer.

- Configure a limit on the number of SA entries that can be stored in cache.

  ```
  configure msdp peer [remoteaddr | all] sa-limit max-sa {vr vr_name}
  ```

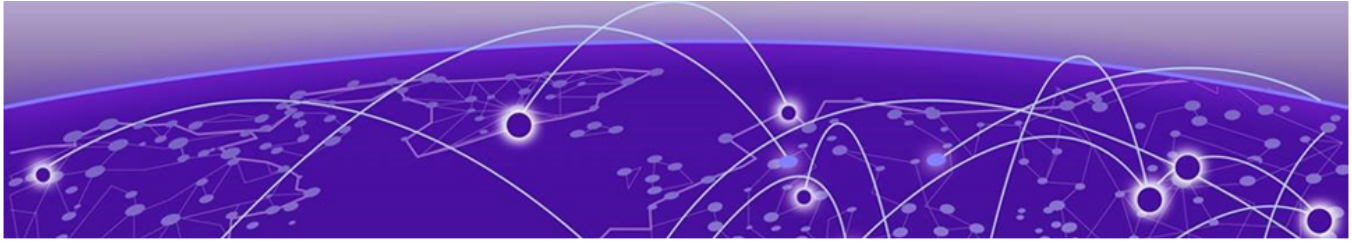
  To allow an unlimited number of SA entries, use 0 (zero) as the value for *max-sa*.

- Display the SA cache limit.

  ```
  show msdp [peer {detail} | {peer} remoteaddr] {vr vr_name}
  ```

SNMP MIBs

SNMP MIB access is not supported for *MSDP*.



Multicast Routing and Switching

Multicast Routing Overview on page 1157
Multicast Table Management on page 1158
PIM Overview on page 1163
IGMP Overview on page 1184
Configuring EAPS Support for Multicast Traffic on page 1190
Configuring IP Multicast Routing on page 1190
Multicast VLAN Registration on page 1199
Displaying Multicast Information on page 1211
Troubleshooting PIM on page 1212

This chapter introduces the features and usage of IP multicasting, which allows a single host on a network to send a packet to a group of hosts. For more information on IP multicasting, refer to the following publications:

- RFC 1112—Host Extension for IP Multicasting
- RFC 2236—Internet Group Management Protocol, Version 2
- RFC 3569—SSM for IPv4/IPv6 (only for IPv4)
- PIM-SM Version 2—draft-ietf-pim-sm--v2-new-05
- RFC 4601—PIM SM (only for IPv4)
- RFC 2362 PIM-SM (Edge Mode)
- RFC 3973 PIM-DM (only for IPv4)
- RFC 3569—draft-ietf-ssm-arch-06.txt PIM-SSM PIM Source Specific Multicast
- PIM-DM Draft IETF Dense Mode—draft-ieft-idmr-pimdm-05.txt, draft-ieft-pim-dm-new-v2-04.txt
- RFC 3376—Internet Group Management Protocol, Version 3

The following URL points to the website for the IETF PIM Working Group: http://datatracker.ietf.org/wg/pim/documents/.

Multicast Routing Overview

Multicast routing and switching is the functionality of a network that allows a single host (the multicast server) to send a packet to a group of hosts. With multicast, the server is not forced to duplicate and send enough packets for all the hosts in a group. Instead, multicast allows the network to duplicate packets for all of the hosts in a group. Multicast greatly reduces the bandwidth required to send data to a group of hosts. IP

multicast routing is a function that allows multicast traffic to be forwarded from one subnet to another across a routing domain.

IP multicast routing requires the following functions:

- A router that can forward IP multicast packets
- A router-to-router multicast routing protocol (for example, Protocol Independent Multicast (PIM)) to discover multicast routes
- A method for the IP host to communicate its multicast group membership to a router (for example, *IGMP (Internet Group Management Protocol)*)

Note
You should configure IP unicast routing before you configure IP multicast routing.

Multicast Table Management

The ExtremeXOS software uses the following tables to support IP multicast traffic:

- IPv4 multicast route table
- L3 hash table
- IP multicast group table
- FDB table (L2 table)
- L2 multicast table (L2MC table)

IP Multicast Hardware Lookup Modes

Extreme platforms support various hardware forwarding lookup modes by using a combination of L3 hash table and L2 (*FDB (forwarding database)*) table. Refer to Multicast Table Management for details on these tables. The scalability limits vary based on the lookup mode used.

Configuration Options

Configuration options allow you to choose the hardware forwarding lookup mode for multicast forwarding. Here is a list of options:

- source-group-vlan -- Uses L3 hash table with S,G,V lookup. This is the default mode for all platforms.
- group-vlan -- Uses L3 hash table with *,G,V lookup.
- mac-vlan -- Uses L2 table with DMAC, *VLAN (Virtual LAN)* lookup.
- mixed-mode -- Uses both L2 and L3 tables for multicast. In this mode, the following logic is applied on installing the cache entries in the hardware:

  Multicast cache entries requiring forwarding across VLANs would be installed in the L3 IP multicast table. This includes PIM, MVR, and PVLAN cache entries.

  Multicast cache entries requiring L2 forwarding within a VLAN are installed in the L2 table. This includes entries corresponding to *IGMP* Snooping, PIM snooping, and MLD snooping.

  Any IPv4/v6 reserved multicast addresses (for example, 224.0.0.x or IPv6 equivalent) are installed in the L3 IP multicast table as needed. These reserved

addresses map to the following multicast MAC addresses: 01:00:5e:00:00:xx, 33:33:00:00:00:xx, 33:33:00:00:01:xx, or 33:33:ff:xx:xx:xx.

Note
Any change in the lookup key configuration causes all cache entries to be cleared, and traffic is temorarily dropped until the system re-learns the multicast caches and associated subscriptions.

Note
mac-vlan mode helps increase scaling. This mode is supported on all platforms.

Note
mac-vlan and mixed-mode are not supported prior to ExtremeXOS 15.3.1.

The ExtremeXOS multicast process continues to maintain the cache entries as "S,G,V", and interacts with HAL the same way as today. ExtremeXOS hardware abstraction layer (HAL) applies the logic explained above and installs the cache entries in the appropriate hardware table. If the cache entry needs to be installed in the L2 multicast table, HAL derives the MAC address based on the standard logic and installs the MAC entry in the L2 table.

The IP multicast address to MAC address mapping is not validated for the received/forwarded multicast packets in ExtremeXOS to date. If the lookup mode is configured either as "mac-vlan" or "mixed-mode", the multicast kernel module is modified to validate this mapping and, if a packet does not use the standard mapping, the packet is dropped.

IPMC Compression

In order to increase the scaling of multicast entries, ExtremeXOS implements a feature called IPMC compression which allows multiple <S,G,V> (or <*,G,V>) IP multicast FDB entries to utilize the same IP multicast group table entry when the associated egress port lists are the same. The base IP multicast compression implementation will be reused for achieving L2 multicast entry reuse. In this case, multiple <MAC,VLAN> multicast FDB entries can use a single L2MC index if the egress port lists of the cache entries are the same.

Interactions with Static FDBs

ExtremeXOS allows you to create FDB entries for multicast MAC address using:

```
create fdb mac_addr {vlan} vlan_name ports port_list
```

These entries also get installed in the L2 table and use the L2MC table for hardware forwarding. If there is a dynamic <MAC,VLAN> entry from MCMGR and a static entry from FDB manager, the static entry takes precedence and the dynamic entry would get deleted in hardware. Compression of L2MC indices is not supported on these types of entries. Each newly created static multicast FDB will cause the allocation of a new L2MC index.

Interactions with Dynamic FDBs

When IP multicast forwarding entries are utilizing the L2 MAC table, the multicast entries are installed as static in the hardware L2 table to avoid undesirable interactions with L2 protocol or user administered FDB flushing. These multicast L2 entries also take precedence over dynamic unicast L2 MAC entries. If there is a hash bucket collision upon inserting an L2 multicast entry, it will replace another dynamic unicast L2 entry if one exists in the same hash bucket.

Virtual Router Support

Current IPMC cache hardware entries stored as <S,G,V> additionally include the VRID associated with the ingress . In this feature, <MAC, VLAN> cache entries are stored in the L2 table which does not additionally include the VRID. However, user VRs are still supported since the VLAN portion of the lookup key is unique across all VRs.

IPMC Cache Rate Limiting

Based on the number of cache entries supported on each platform, there is a software cache limiting implementation present in ExtremeXOS multicast. The HAL module informs MCMGR about the supported limit, MCMGR creates cache entries up to MAX supported limit, and the remaining traffic is dropped in software.

Flood to VLAN Filters for mDNS, LLMNR, and UPnP protocols

Version 32.2 introduces support for flood to VLAN Filters for mDNS, LLMNR, and UPnP protocols. This feature enables you to create filters that forward to VLAN for these three protocol packets on both standalone switches and stacks. When flood to VLAN is enabled, an ACL filter is installed, and one ACL entry is consumed for each of the three protocols. When in learn mode, no ACL filters are installed, and no ACL resources are consumed. The default mode is learn.

Supported Platforms

This feature is implemented on all platforms.

Limitations

The following limitations exist for the L2MC table feature:

- When using the "mac-vlan" configuration option:

  PIMv4/V6, MVR features cannot be used.

  IGMPv3 should not be used in conjunction with this mode

  Private VLAN multicast should not be used.

  Issues with IP multicast address to MAC address mapping:

  All IPv4 multicast frames use multicast mac addresses starting with 01:00:5e:xx:xx:xx. The lower order 23 bits of the IP multicast address is used in the MAC address derivation. As only 23 bits of MAC addresses are available for mapping layer 3 IP multicast addresses to layer 2 MAC addresses, this mapping results in 32:1 address ambiguity.

  When traffic is received for 1 out of these 32 overlapping address, then the MAC, Vlan entry is installed in hardware based on the IGMP group membership of received

traffic's destination multicast IP address. After this installation, traffic to any of the remaining 31 addresses is delivered based on the existing cache entry and the actual receiver list of the remaining 31 addresses will not be honored.

IPv6 multicast streams use multicast MAC addresses in the form 33:33:xx:xx:xx:xx. The lower 24 bits of the IPv6 multicast address are used to derive the MAC address. So, the address ambiguity issue is also applicable to IPv6 with more severity. Given this condition, we do not recommend using overlapping IP multicast addresses with this mode.

This limitation applies to "mixed-mode", too.

- IPv4 multicast addresses consist of a block of addresses (224.0.0.x) used for network control traffic. Packets having IP destination addresses from the LNCB are always flooded to all ports of the VLAN. The address ambiguity issue discussed above is applicable for the addresses in this block too. For example, 224.0.0.5 (address used for *OSPF (Open Shortest Path First)*) and 225.0.0.5 would use the same MAC address 01:00:5e:00:00:05. If a mac based multicast FDB entry is installed on the hardware for 01:00:5e:00:00:05 based on the 225.0.0.5 join list, it would break OSPF functionality. Hence, MAC addresses mapping to the LNCB block will not be installed in the L2 table, resulting in software forwarding for those streams. We recommend that you avoid useing multicast addresses that map to the 01:00:5e:00:00:xx MAC address range.
- As per RFC 3307, IANA assigned reserved IPv6 multicast addresses could be in the group Id range of 0x00000001 to 0x3FFFFFFF. As a result, ExtremeXOS switches flood traffic addressed to ff02::/98 to all ports of the VLAN. Since the lower 32 bits of IPv6 multicast addresses are mapped to the multicast mac address, not installing all of the addresses in this range would make it too restrictive. So, installing entries for 33:33:00:00:00:xx in hardware would be avoided, and the traffic would be software forwarded.

  In addition, the following important IPv6 multicast addresses cannot be installed as hardware forwarding entries:

  DHCP (Dynamic Host Configuration Protocol): All-dhcp-agents address FF02:0:0:0:0:0:1:2

  All-dhcp-servers address FF05:0:0:0:0:0:1:3

  Neighbor Discovery (ND): Solicited-node-address FF02::1:FF00:0000/104

  Therefore, the following multicast MAC addresses are not programmed in hardware, and the corresponding packets are handled in slowpath: 33:33:00:00:01:xx , 33:33:ff:xx:xx:xx
- Given the issues with IP multicast address to MAC address mapping, no attempt is made to merge subscriber lists of multiple overlapping IP groups.
- The following limitation regarding IPMC compression is also applicable for this feature, because this feature uses the same L2MC entry for multiple l2 multicast entries with same egress ports.

- When IP multicast forwarding entries are installed as <MAC,VLAN>, IGMP or MLD packets which have a MAC-DA=<group> will cause the refresh of the IP multicast cache, preventing timely entry age-out.
- The L2MC table is limited to 1K entries on all platforms. This means that only up to 1K unique port lists can be addressed from the <MAC,VLAN> IP multicast forwarding entries that are stored in the L2 table. Additionally, statically created multicast FDB entries do not perform L2MC index compression.

IPv4 Multicast Route Table

Beginning with Release 12.1, all IP multicast routes are stored and maintained in the software multicast route table. Routes are added to the multicast route table from the following sources:

- Multicast static routes (configured manually by the network administrator)
- Multicast dynamic routes (learned through protocols such as MBGP and MISIS)

The multicast route table is used for reverse path forwarding (RPF) checks, not for packet forwarding. The switch uses RPF checks to avoid multicast forwarding loops. When a multicast packet is received, the switch does an RPF lookup, which checks the routing tables to see if the packet arrived on the interface on which the router would send a packet to the source. The switch forwards only those packets that arrive on the interface specified for the source in the routing tables.

The RPF lookup uses the multicast routing table first, and if no entry is found for the source IP address, the RPF lookup uses the unicast routing table.

Note

Because the multicast routing table is used only for RPF checks (and not for routing), IP route compression and *ECMP (Equal Cost Multi Paths)* do not apply to multicast routes in the multicast routing table.

Beginning with ExtremeXOS software version 12.1, the route metric is no longer used to select between multicast and unicast routes. If the RPF lookup finds a route in the multicast table, that route is used. The unicast routing table is used only when no route is found in the multicast table.

The advantage to having separate routing tables for unicast and multicast traffic is that the two types of traffic can be separated, using different paths through the network.

L3 Hash Table

The L3 hash table is introduced in Introduction to Hardware Forwarding Tables on page 897. The L3 hash table stores entries for IPv4 routes, IPv4 and IPv6 hosts, and IPv4 and IPv6 multicast groups. For multicast, L3 hash table supports <S,G,V> and <*,G,V> lookups. The entry from this table provides an index to IP Multicast Group table.

To make more space available in the L3 hash table for IPv4 and IPv6 multicast groups, you can do the following:

- Configure the extended IPv4 host cache feature to move IPv4 local and remote routes to the LPM table as described in Extended IPv4 Host Cache on page 897.
- For more information, see the description for the **`configure iproute reserved-entries`** command.

IP Multicast Group Table

The IP multicast group table specifies the egress ports for Layer 2 and Layer 3 multicast traffic groups. To make more space available in the IP multicast group table, you can Leave IP multicast compression enabled (it is enabled by default). This allows multiple L3 hash entries to share the same IP multicast group entry if the egress list is the same.

PIM Overview

Protocol Independent Multicast (PIM) is the de-facto standard for routing multicast traffic over the Internet. Other multicast routing protocols such as DVMRP and MOSPF are sometimes used in controlled environment, but are not widely deployed. PIM does not depend on a particular unicast routing protocol for its operation. Also, it does not have any mechanism of its own for route discovery. PIM operation is based on the routing table being populated by another routing protocol, or by the user. This provides flexibility in routing unicast and multicast traffic based on a common database.

PIM has two flavors, sparse and dense mode, that are deployed in different topologies. These two flavors, called PIM-SM and PIM-DM, are different in operation. PIM-SM is based on a "join protocol", where traffic is not forwarded on a segment unless an explicit request originates (typically through *ICMP*) from the network segment. PIM-DM is based on a "flood and prune" mechanism, where every one receives traffic until they explicitly inform (through the PIM-DM prune mechanism) that they do not want that particular stream. Thus, PIM-DM is typically deployed in topologies where listeners are densely populated. And PIM-SM is typically deployed where the receivers are sparsely populated over the network, so that most of the network segments' bandwidth is conserved.

You can configure dense mode or sparse mode on a per-interface basis. After they are enabled, some interfaces can run dense mode, while others run sparse mode. The switch supports both dense mode and sparse mode operation.

The switch also supports PIM snooping.

PIM Edge Mode

PIM Edge Mode is a subset of PIM that operates with the following restrictions:

- The switch does not act as a candidate rendezvous point (CRP).
- The switch does not act as a candidate bootstrap router (CBSR).

- At most, four active PIM-SM interfaces are permitted. There is no restriction on the number of passive interfaces (within the limit of the maximum IP interfaces).
- Only PIM Sparse Mode (PIM-SM) is supported.

Note
This feature is supported at and above the license level listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

Active PIM interfaces can have other PIM enabled routers on them. Passive interfaces should only have hosts sourcing or receiving multicast traffic. If another PIM router is connected to a multi-access VLAN, then passive mode should not be enabled for that respective VLAN. *OSPF* passive mode should not be enabled for a VLAN when a PIM neighbor is present.

PIM Dense Mode

Protocol-Independent Multicast - Dense Mode (PIM-DM) is a multicast routing protocol. PIM dense-mode is a flood and prune-based protocol. Convergence is based on the downstream routers' response for the traffic received. The downstream router in turn floods the traffic to its own downstream interfaces. Each router sends prune to the interface on which it received the traffic under the following conditions:

- Traffic was not received on RPF interface towards the source.
- The PIM router is a leaf router, and there are no *IGMP*/MLD members.
- All the downstream PIM routers have pruned the stream, and there are no IGMP/MLD members.

A feature called PIM-DM state refresh creates two PIM-DM operating modes, which are described in the following sections:

Note
For additional information on PIM-DM, see RFC 3973, Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification.

PIM-DM Without State Refresh

PIM-DM is a broadcast and prune protocol, which means that multicast servers initially broadcast traffic to all destinations, and then switches later prune paths on which there are no receivers. Figure 141 shows a dense mode multicast tree with an active branch and a pruned branch.

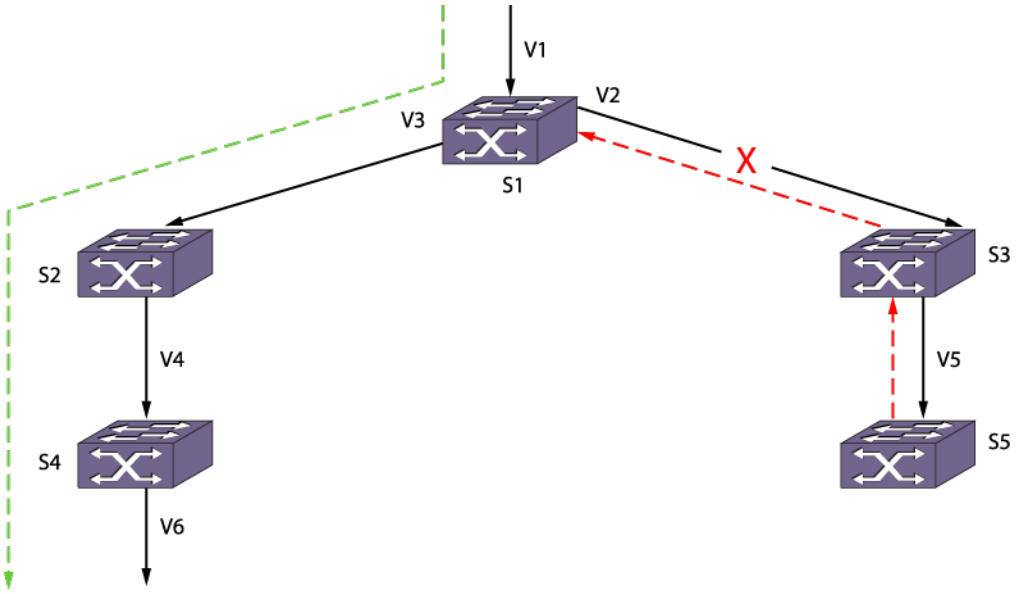


Figure 141: PIM-DM Operation

Multicast traffic is flowing from *VLAN* V1 connected to switch S1. S1 floods multicast traffic to both neighbors S2 and S3 which in turn flood multicast traffic to S4 and S5. S4 has *IGMP* members, so it floods multicast traffic down to VLAN V6. S5, which has no multicast members, sends a prune upstream towards the source. The green line shows the flow of traffic on the active branch, and the red line shows the prune sent upstream for the pruned branch. After outgoing interface V2 is pruned from the multicast tree, subsequent multicast traffic from S1 flows only through S2 and S4 and is not forwarded to S3.

After S3 sends a prune upstream, S3 starts a prune hold time timer on outgoing interface V5. When this timer expires, S3 adds V5 back to the multicast egress list and sends a graft upstream to pull multicast traffic down again. When multicast traffic arrives from S1, it is forwarded to S5, which repeats the upstream prune message because it still has no members. This prune, time-out, and flood process repeats as long as the traffic flow exists and no members are on the pruned branch, and this process consumes bandwidth during every cycle.

Note
This feature is supported at and above the license level listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

PIM-DM routers perform reverse path multicasting (RPM). However, instead of exchanging its own unicast route tables for the RPM algorithm, PIM-DM uses the existing unicast routing table for the reverse path. As a result, PIM-DM requires less system memory.

PIM-DM with State Refresh

The PIM-DM State Refresh feature keeps the PIM-DM prune state from timing out by periodically sending a state refresh control message down the source tree. These control messages reset the prune hold time timer on each pruned interface and prevent the bandwidth waste that occurs with each prune, time-out, and flood cycle.

When a topology change occurs, the PIM-DM State Refresh feature improves network convergence. For example, suppose that an S, G entry on S5 in the following figure is removed due to non-availability of a route. Without PIM-DM State Refresh, multicast traffic is blocked for minutes (due to a time-out on the upstream routers). In the meantime if an *IGMP* member or a PIM-DM neighbor joins S5, there is no way to pull traffic down immediately because S5 does not have any S, G information. State refresh control messages solve this problem by indicating S, G state information periodically to all downstream routers. When S5 receives a state refresh from S3, it scans the S, G information and all pending requests from PIM-DM neighbors and IGMP members. If there are pending requests for the group in the state refresh message, S5 can immediately send a graft message upstream to circumvent the upstream timers and pull multicast traffic to its members and neighbors.

- To enable, configure, and disable the PIM-DM State Refresh feature, use the following commands:

  ```
  configure pim state-refresh {vlan} [vlanname | all] [on | off]
  configure pim state-refresh timer origination-interval interval
  configure pim state-refresh timer source-active-timer interval
  configure pim state-refresh ttl ttlvalue
  ```

PIM Sparse Mode

Unlike PIM-DM, Protocol-Independent Multicast - Sparse Mode (PIM-SM) is an explicit join protocol, which means that multicast receivers, and the routers that support them, must join multicast groups before they receive multicast traffic. When all receivers on a network branch leave a multicast group, that branch is pruned so that the multicast traffic does not continue to consume bandwidth on that branch. PIM-SM supports shared trees as well as shortest path trees (SPTs). PIM-SM is beneficial for large networks that have group members that are sparsely distributed.

Note
This feature is supported at and above the license level listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

Using PIM-SM, the router sends a join message to the rendezvous point (RP). The RP is a central multicast router that is responsible for receiving and distributing the initial multicast packets. You can configure a dynamic or static RP.

When a router has a multicast packet to distribute, it encapsulates the packet in a unicast message and sends it to the RP. The RP decapsulates the multicast packet and distributes it among all member routers.

When a router determines that the multicast rate has exceeded a configured threshold, that router can send an explicit join to the originating router. When this occurs, the receiving router gets the multicast directly from the sending router and bypasses the RP.

Note
You can run either PIM-DM or PIM-SM per *VLAN*.

PIM Mode Interoperation

An Extreme Networks switch can function as a PIM multicast border router (PMBR). A PMBR integrates PIM-SM and PIM-DM traffic.

When forwarding PIM-DM traffic into a PIM-SM network, the PMBR acts as a virtual first hop and encapsulates the initial traffic for the RP. The PMBR forwards PIM-DM multicast packets to the RP, which, in turn, forwards the packets to those routers that have joined the multicast group.

The PMBR also forwards PIM-SM traffic to a PIM-DM network, based on the (*.*.RP) entry. The PMBR sends a (*.*.RP) join message to the RP, and the PMBR forwards traffic from the RP into the PIM-DM network.

No commands are required to enable PIM mode interoperation. PIM mode interoperation is automatically enabled when a dense mode interface and a sparse mode interface are enabled on the same switch.

PIM Source Specific Multicast

PIM-SM works well in many-to-many multicasting situations. For example, in video conferencing, each participating site multicasts a stream that is sent to all the other participating sites. However, PIM-SM is overly complex for one-to-many multicast situations, such as multimedia content distribution or streaming stock quotes. In these and similar applications, the listener is silent and can know the source of the multicast in advance, or can obtain it. In these situations, there is no need to join an RP, as the join request can be made directly towards the source.

Note
This feature is supported at and above the license level listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

Note
(*;G)s are created for groups inside the SSM range if SSM is not enabled.

PIM Source Specific Multicast (PIM-SSM) is a special case of PIM-SM, in which a host explicitly sends a request to receive a stream from a specific source, rather than from any source.

IGMPv3 hosts can use PIM SSM directly, because the ability to request a stream from a specific source first became available with IGMPv3. The PIM-SSM capable router interprets the IGMPv3 message to initiate a PIM-SM join towards the source.

Note
IGMPv1 and IGMPv2 hosts can use PIM SSM if *IGMP*-SSM mapping is enabled and configured on the ExtremeXOS switch. For more information, see Using IGMP-SSM Mapping.

Table 79 describes PIM-SSM behavior while sending IGMPV3 joins in the SSM range and outside the SSM range for IPv4:

Table 79: Using PIM-SSM While Sending IGMPV3 Joins (IPv4)

				ExtremeXOS 15.4		ExtremeXOS 15.5	
SSM Enabled	SSM range	Mode	Include Src	Action	Observation	Action	Observation
No	Yes	Incl	Yes	Send IGMPv3 join in SSM range	-the group is learned -(*;G) is not created	Send IGMPv3 join in SSM range	-the group is learned - (*;G) is created
No	Yes	Incl	Yes	Send IGMPv3 out of SSM range	-the group is learned -no (*;G) is created	Send IGMPv3 out of SSM range	-the group is learned - (*;G) is created
No	Yes	Excl	No	Send IGMPv3 join in SSM range	-the group is not learned (PD4-3138792131) -no (*;G) is created	Send IGMPv3 join in SSM range	-the group is not learned (PD4-3138792131) -no (*;G) is created
No	Yes	Excl	No	Send IGMPv3 join out of SSM range	-the group is learned - (*;G) is created	Send IGMPv3 join out of SSM range	-the group is learned - (*;G) is created
No	Yes	Excl	Yes	Send IGMPv3 join in SSM range	-the group is not learned -no (*;G) is created	Send IGMPv3 join in SSM range	-the group is not learned -no (*;G) is created
No	Yes	Excl	Yes	Send IGMPv3 join out SSM range	-the group is learned -no (*;G) is created	Send IGMPv3 join out SSM range	-the group is learned -no (*;G) is created
No	No	Incl	Yes	Send IGMPv3 join	-the group is learned -no (*;G) is created	Send IGMPv3 join	-the group is learned -(*;G) is created
No	No	Excl	No	Send IGMPv3 join	-the group is learnt -(*;G) is created	Send IGMPv3 join	-the group is learnt -(*;G) is created
No	No	Excl	Yes	Send IGMPv3 join	-the group is learned -no (*;G) is created	Send IGMPv3 join	-the group is learned -no (*;G) is created

Table 79: Using PIM-SSM While Sending IGMPV3 Joins (IPv4) (continued)

				ExtremeXOS 15.4		ExtremeXOS 15.5	
SSM Enabled	SSM range	Mode	Include Src	Action	Observation	Action	Observation
Yes	Yes	Incl	Yes	Send IGMPv3 join in SSM range	-the group is learned -(S;G) is created	Send IGMPv3 join in SSM range	-the group is learned -(S;G) is created
Yes	Yes	Incl	Yes	Send IGMPv3 out of SSM range	-the group is learned -no (*;G) is created	Send IGMPv3 out of SSM range	-the group is learned - (*;G) is created
Yes	Yes	Excl	No	Send IGMPv3 join in SSM range	-the group is not learned -no (*;G) is created	Send IGMPv3 join in SSM range	-the group is not learned -no (*;G) is created
Yes	Yes	Excl	No	Send IGMPv3 join out of SSM range	-the group is learned - (*;G) is created	Send IGMPv3 join out of SSM range	-the group is learned - (*;G) is created
Yes	Yes	Excl	Yes	Send IGMPv3 join in SSM range	-the group is not learned -no (*;G) is created	Send IGMPv3 join in SSM range	-the group is not learned -no (*;G) is created
Yes	Yes	Excl	Yes	Send IGMPv3 join out SSM range	-the group is learned -no (*;G) is created	Send IGMPv3 join out SSM range	-the group is learned -no (*;G) is created
Yes	No	Incl	Yes	Send IGMPv3 join	-the group is learned -no (*;G) is created	Send IGMPv3 join	-the group is learned -(*;G) is created
Yes	No	Excl	No	Send IGMPv3 join	-the group is learned - (*;G) is created	Send IGMPv3 join	-the group is learned - (*;G) is created
Yes	No	Excl	Yes	Send IGMPv3 join	-the group is learned -no (*;G) is created	Send IGMPv3 join	-the group is learned -no (*;G) is created

Table 80 describes PIM-SSM behavior while sending MLDV2 joins in the SSM range and outside the SSM range for IPv6:

Table 80: Using PIM-SSM While Sending MLDV2 Joins (IPv6)

				ExtremeXOS 15.4		ExtremeXOS 15.5	
SSM Enabled	SSM range	Mode	Include Src	Action	Observation	Action	Observation
No	Yes	Incl	Yes	Send MLDv2 join in SSM range	-the group is learned - no (*;G) is created	Send MLDv2 join in SSM range	-the group is learned -(*;G) is created
No	Yes	Incl	Yes	Send MLDv2 out of SSM range	-the group is learned - no (*;G) is created - (S;G) is created	Send MLDv2 out of SSM range	-the group is learned -(*;G) is created
No	Yes	Excl	No	Send MLDv2 join in SSM range	-the group is learned -(*;G) is created	Send MLDv2 join in SSM range	-the group is not learned - no (*;G) is created
No	Yes	Excl	No	Send MLDv2 join out of SSM range	-the group is learned -(*;G) is created	Send MLDv2 join out of SSM range	-the group is learned -(*;G) is created
No	Yes	Excl	Yes	Send MLDv2 join in SSM range	-the group is learned -(*;G) is created	Send MLDv2 join in SSM range	-the group is not learned - no (*;G) is created
No	Yes	Excl	Yes	Send MLDv2 join out SSM range	-the group is learned -(*;G) is created	Send MLDv2 join out SSM range	-the group is learned -(*;G) is created
No	No	Incl	Yes	Send MLDv2 join	-the group is learned - (S;G) is created	Send MLDv2 join	-the group is learned - (*;G) is created
No	No	Excl	No	Send MLDv2 join	-the group is learned -(*;G) is created	Send MLDv2 join	-the group is learned -(*;G) is created
No	No	Excl	Yes	Send MLDv2 join	-the group is learned -(*;G) is created	Send MLDv2 join	-the group is learned -(*;G) is created

Table 80: Using PIM-SSM While Sending MLDV2 Joins (IPv6) (continued)

				ExtremeXOS 15.4		ExtremeXOS 15.5	
SSM Enabled	SSM range	Mode	Include Src	Action	Observation	Action	Observation
Yes	Yes	Incl	Yes	Send MLDv2 join in SSM range	-the group is learned - (S;G) is created	Send MLDv2 join in SSM range	-the group is learned - (S;G) is created
Yes	Yes	Incl	Yes	Send MLDv2 out of SSM range	-the group is learned - no (*;G) is created - (S;G) is created	Send MLDv2 out of SSM range	-the group is learned -(*;G) is created
Yes	Yes	Excl	No	Send MLDv2 join in SSM range	-the group is learned -(*;G) is created	Send MLDv2 join in SSM range	-the group is not learned - no (*;G) is created
Yes	Yes	Excl	No	Send MLDv2 join out of SSM range	-the group is learned -(*;G) is created	Send MLDv2 join out of SSM range	-the group is learned -(*;G) is created
Yes	Yes	Excl	Yes	Send MLDv2 join in SSM range	-the group is learned -(*;G) is created	Send MLDv2 join in SSM range	-the group is not learned - no (*;G) is created
Yes	Yes	Excl	Yes	Send MLDv2 join out SSM range	-the group is learned -(*;G) is created	Send MLDv2 join out SSM range	-the group is learned -(*;G) is created
Yes	No	Incl	Yes	Send MLDv2 join	-the group is learned - (S;G) is created	Send MLDv2 join	-the group is learned - (*;G) is created
Yes	No	Excl	No	Send MLDv2 join	-the group is learned -(*;G) is created	Send MLDv2 join	-the group is learned -(*;G) is created
Yes	No	Excl	Yes	Send MLDv2 join	-the group is learned -(*;G) is created	Send MLDv2 join	-the group is learned -(*;G) is created

PIM-SSM has the following advantages:

- No overhead of switching to the source-specific tree and waiting for the first packet to arrive
- No need to learn and maintain an RP
- Fewer states to maintain on each router
- No need for the complex register mechanism from the source to the RP
- Better security, as each stream is forwarded from sources known in advance

PIM-SSM has the following requirements:

- Any host that participates directly in PIM-SSM must use IGMPv3. For PIM IPv6 host must use MLDv2.
- To support IGMPv1 and IGMPv2 hosts, IGMP-SSM mapping must be enabled and configured. To support MLDv1 hosts, MLD-SSM mapping must be enabled and configured.

PIM-SSM is designed as a subset of PIM-SM and all messages are compliant with PIM-SM. PIM-SSM and PIM-SM can coexist in a PIM network; only the last hop router need to be configured for PIM-SSM if both source and receivers are present all the time. However, to avoid any JOIN delay, it is recommended that you enable all routers along the (s,g) path for PIM-SSM.

Configuring the PIM-SSM Address Range

A range of multicast addresses is used for PIM-SSM. Within that address range, non-IGMPv3 messages are ignored, and any IGMPv3 exclude messages are ignored. These messages are ignored for all router interfaces, even those not configured for PIM-SSM. By default there is no PIM-SSM range specified on the router. If you choose the default keyword in the CLI when specifying the PIM-SSM range, you configure the range 232.0.0.0/8 and default PIM-SSM range for IPv6 is FF3x::/96. You can also choose to specify a different range for PIM-SSM by using a policy file.

To configure the PIM-SSM address range, use the following command:

```
configure pim ssm range [default | policy policy-name]
```

PIM Snooping

PIM snooping provides a solution for handling multicast traffic on a shared media network more efficiently. In networks where routers are connected to a L2 switch, multicast traffic is essentially treated as broadcast traffic (see Figure 142).

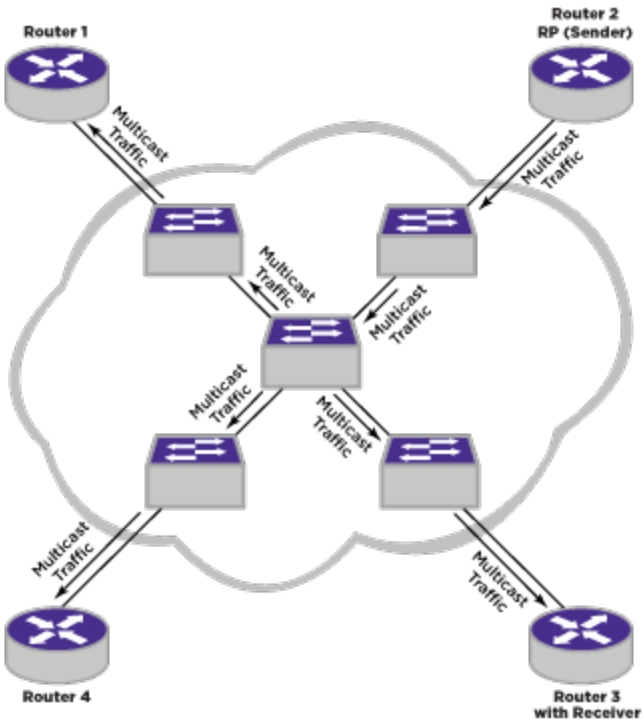


Figure 142: Multicast Without PIM Snooping

IGMP snooping does not solve this flooding issue when routers are connected to a L2 switch. Switch ports are flooded with multicast packets. PIM snooping addresses this flooding behavior by efficiently replicating multicast traffic only onto ports which routers advertise the PIM join requests (see Figure 143).

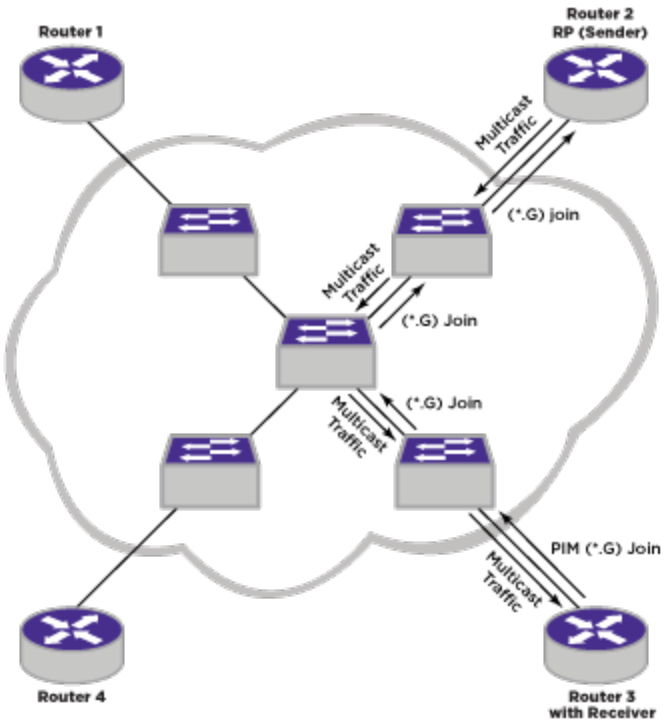


Figure 143: Multicast With PIM Snooping

PIM snooping does not require PIM to be enabled. However, IGMP snooping must be disabled on *VLAN*s that use PIM snooping. PIM snooping and MVR cannot be enabled on the same VLAN.

To enable PIM snooping on one or all VLANs, use the following command:

```
enable pim snooping {{vlan} name}
```

To disable PIM snooping on one or all VLANs, use the following command:

```
disable pim snooping {{vlan} name}
```

Note
PIM snooping can be enabled only between PIM SM enabled switches. It should not enabled between PIM DM enabled switches.

PIM Register Policy

This feature allows you to filter register messages based on the policy file configured at the First Hop Router (FHR) and Rendezvous Point (RP) in PIM-SM domain. You can use the register policy to filter out specific PIM register messages that have encapsulated specific (S,G) packets. This feature allows you to detect and deny malicious multicast packets from flowing into a multicast shared tree, and causing a potential service blackout. The PIM Register Policy feature is supported in both the PIM IPV4 and PIM IPV6 mode .

Filtering at FHR

- FHR receives the source multicast packet and sends a register message towards the RP. Before it sends the register message to the RP, the FHR checks the configured register filter policy. If the (S,G) is denied by the policy, the register will not send a message to the RP. The FHR adds the L3 entries to stop the packet from arriving at the CPU. An EMS message is logged.
- The FHR checks the register policy before generating a NULL register packet. If the policy is denied by the filter then the NULL register is not sent to the RP.
- If the cache's Group is in the SSM range, or is received in the PIM dense circuit, then this filtering is not applicable. The cache miss packet will go thru the normal processing.
- If a non-SSM (S,G) cache already exists but is denied by the filter policy, then (S,G) cache is removed. The cache miss comes to the CPU for register processing if the traffic is still flowing.

The PIM filtering policy is configured at the FHR using the `configure pim {`**`ipv4`** `|` **`ipv6`**`} register-policy [`**`policy`** `|` **`none`**`]` command.

Filtering at RP

- When an encapsulated PIM register packet or PIM NULL register is received by the RP, and is denied by the registering filter policy, the register message is discarded. Additionally, no (S,G) cache is created in the PIM cache.
- The register drop counter is incremented, and the EMS message is logged.
- If a register is received from the *MSDP (Multicast Source Discovery Protocol)*, it also goes through the RP filtering policy.

The PIM filtering policy is configured at RP using the following command:

`configure pim {`**`ipv4`** `|` **`ipv6`**`} register-policy rp [`**`rp_policy_name`** `|` **`none`**`]`

Example

PIM Register policy rules can be created only on the Source IP address and group address, as shown in the following.

SAMPLE POLICY 1:

If the matching entry is not found, then the entry permission is considered as denied. Need wildcard entries to permit the unmatched entries. The following policy file permits the matched entries and wildcard entries deny the unmatched source:

```
entry policy1 {
     if match any{
         nlri 10.10.10.0/24;
         nlri 20.20.20.0/24 ;
      }                        <= Permit these sources for group
      then {
          nlri 226.1.1.0/24;
          nlri 227.1.1.0/24;
          permit;
      }
}
```

```
entry policy2 {
     if match any{
         nlri 10.10.10.0/24;
      }
      then {
          nlri 228.1.1.0/24;
          permit;
      }
}

entry policy3 {
     if match any{
      }
      then {                          <= deny all other source;
             deny;
      }
}
```

SAMPLE POLICY 2:

The following policy file denies the matched entries and wildcard entry permits the unmatched source:

```
entry policy1 {
     if match any{
         nlri 10.10.10.0/24;
         nlri 20.20.20.0/24 ;
      }                       <= deny these sources for group
      then {
          nlri 226.1.1.0/24;
          nlri 227.1.1.0/24;
          deny;
      }
}

entry policy2 {
     if match any{
         nlri 10.10.10.0/24;
      }
      then {
          nlri 228.1.1.0/24;
          deny;
      }
}

entry policy3 {
     if match any{
      }
      then {                          <= permit all other source;
             permit;
      }
}
```

SAMPLE POLICY 3:

The following policy file denies the matched entries and unmatched entries. Since the wildcard permit entry is not present in this policy file, by default all the unmatched entries are denied:

```
entry policy1 {
     if match any{
         nlri 10.10.10.0/24;
```

```
         nlri 20.20.20.0/24 ;
      }                        <= deny these sources for group
      then {
          nlri 226.1.1.0/24;
          nlri 227.1.1.0/24;
          deny;
      }
}

entry policy2 {
     if match any{
         nlri 10.10.10.0/24;
      }
      then {
          nlri 228.1.1.0/24;
          deny;
      }
}
```

PIM DR Priority

The DR_Priority option allows a network administrator to give preference to a particular router in the DR election process by giving it a numerically larger DR Priority. The DR_Priority option is included in every Hello message, even if no DR Priority is explicitly configured on that interface. This is necessary because priority-based DR election is only enabled when all neighbors on an interface advertise that they are capable of using the DR_Priority Option. The default priority is 1.

DR Priority is a 32-bit unsigned number, and the numerically larger priority is always preferred. A router's idea of the current DR on an interface can change when a PIM Hello message is received, when a neighbor times out, or when a router's own DR Priority changes. If the router becomes the DR or ceases to be the DR, this will normally cause the DR Register state machine to change state. Subsequent actions are determined by that state machine.

The DR election process on the interface consists of the following:

- If any one of the neighbors on the interface is not advertised, the DR priority (not DR capable) will not considered for the all the neighbors in the circuit and the primary IP address will be considered for all the neighbors.
- Higher DR priority or higher primary address will be elected as DR.

Use the following command to configure PIM DR Priority:

```
configure pim {ipv4 | ipv6} {vlan} [vlan_name] dr-priority priority
```

PIM ECMP Load Splitting

The PIM *ECMP* feature allows downstream PIM routers to choose multiple ECMP paths to source via hash from one of following selections without affecting existing unicast routing algorithm:

- Source
- Group

- Source-Group
- Source-Group-Next Hop

This feature operates on a per (S,G) basis splitting the load onto available equal-cost paths by hashing according to the selection criteria configured by the user. It does not operate by counting the flows. Load splitting need not balance the traffic on the available paths. PIM ECMP load splitting uses a hash algorithm based on the selected criteria to pick up the path to use and will result in load-sharing the traffic when there are many multicast streams that utilize approximately the same amount of bandwidth.

PIM ECMP Load Splitting Based on Source Address

When you enable PIM ECMP load splitting based on source address, the RPF interface for each (*, G) or (S,G) state is selected among the equal cost paths based on the hash derived from the source address. For an (S, G) state, the address considered for hashing is the source address of the state. For a (*, G) state, the address considered for hashing is the address of the RP that is associated with the state's group address. There is no randomization applied when calculating the hash value. The same hash value is generated on all the ExtremeXOS routers for a given source address. If there are two equal cost paths ("left" and "right") available at the last hop router and at each of the intermediate routers for a given source, each of these routers pick the same hash, and the traffic flows can get skewed (to either "left" or "right" paths).

PIM ECMP Load Splitting Based on Group Address

When you enable PIM ECMP load splitting based on group address, the RPF interface for each (*, G) or (S,G) state is selected among the equal cost paths based on the hash derived from the group address. If multiple equal cost common paths exist to the multicast source and the RP that is associated with the state's group address, the same hash will be chosen for both (*, G) and (S, G) states as the same group address is used in deriving the hash. There is no randomization applied when calculating the hash value. The same hash value is generated on all the ExtremeXOS routers for a given group address. If there are two equal cost paths ("left" and "right") available at the last hop router and at each of the intermediate routers for a given group, each of these routers pick the same hash and the traffic flows can get skewed (to either "left" or "right" paths).

PIM ECMP Load Splitting Based on Source-Group Addresses

When you enable PIM ECMP load splitting based on source-group address, the RPF interface for each (*, G) or (S,G) state is selected among the equal cost paths based on the hash derived from the source and group addresses. For an (S, G) state, the address considered for hashing is the source address of the state. For a (*, G) state, the address considered for hashing is the address of the RP that is associated with the state's group address. There is no randomization applied when calculating the hash value. The same hash value is generated on all the ExtremeXOS routers for a given source-group address. If there are two equal cost paths ("left" and "right") available at the last hop router and at each of the intermediate routers for a given source-group, each of these routers pick the same hash and the traffic flows can get skewed (to either "left" or "right" paths).

PIM ECMP Load Splitting Based on Source-Group-Next Hop Addresses

When you enable PIM ECMP load splitting based on source-group-next hop address, the RPF interface for each (*, G) or (S,G) state is selected among the equal cost paths based on the hash derived from the source, group and next hop addresses. The hash value derived after introducing the next hop address is still predictable as there is no randomization applied when calculating the hash value. However, since the next hop address used at each of the routers vary, the hash value generated on each of the ExtremeXOS routers is different. As the hash value is different on each of the routers, the problem of traffic path skew present in the above mentioned schemes does not exist in this scheme.

Reconvergence Due to Unicast Routing Changes

When a unicast route to a source or RP address changes (when a path goes down or a new path becomes available), all the (*, G) and (S, G) states change based on the available unicast route information provided by Route Manager process. If one of the paths goes down and comes back up, multicast forwarding will reconverge to same RPF path that was used before the path went down. The hash function based on Source-Group-Next Hop avoids skewing of traffic flows because it introduces the actual next-hop IP address of PIM neighbors into the calculation resulting in different hash value being computed for each router. The Source-Group-Next Hop based hash function doesn't take the total number of available paths into consideration and so it increases stability of the paths chosen during path failures. During path failures, the multicast states that were using the failed path would need to reconverge onto the remaining paths. All other states using the unaffected paths are not affected.

Limitations

- Cannot be used along with static multicast routes.
- Not supported for ExtremeXOS Multicast Tools (mtrace and mrinfo) in current release.
- Load splitting is not applied for configured static multicast routes and multicast routes present in the multicast routing table.
- Load splitting is only effective when the equal cost paths are upstream PIM neighbors on different interfaces. When the equal cost paths are PIM neighbors on the same shared *VLAN*, PIM assert mechanism chooses one path to avoid traffic duplication. The path chosen by PIM assert mechanism overrides the path selected by Multicast ECMP load splitting.

IPv6 Specific Features

Apart from adding support for IPv6 addresses, PIMv6 adds the following functionality to existing PIM implementation:

- Secondary address list - This is a new option which will be added to the V6 Hello messages sent. The list includes all addresses assigned to an interface, including the link local addresses. The receiving router must process these addresses and must associate the same with the neighbor that sent the message.
- Tunnel interface - This is similar to a *VLAN* interface. APIs are now added to get callbacks from VLAN manager client for IP address configuration for a tunnel, etc.

Secondary address list

The Address List Option, in a Hello message, advertises all the secondary addresses associated with the source interface of the router originating the message. These addresses are associated with the neighbor, and are used to compute the neighbor's primary address. The function NBR uses information gathered through PIM Hello messages to map the IP address A of a directly connected PIM neighbor on interface I to the primary IP address of the same router. The primary IP address of a neighbor is the address that it uses as the source of its PIM Hello messages.

Tunnel interfaces

Two PIMv6 domains can be connected through an IPv4 network. In this case, PIMv6 routers across the domains communicate over the IPv4 network by tunneling the IPv6 packets inside IPv4 headers. To enable such communication, PIMv6 provides support for Tunnel interfaces.

The following tunnel types are supported:

- 6-in-4
- 6-to-4

Configuration details

PIMv6 is incorporated into all CLI commands that currently support the PIM implementation. New keywords are added to support IPv6, and show command output is modified to display IPv6 related information. For specific configuration details, refer to the Switch Engine v33.6.1 Command References .

Anycast Rendezvous Point (RP) Using PIM (RFC 4610)

Anycast Rendezvous Point (RP) Using PIM (*RFC 4610*) provides fast convergence when RP routers fail using PIM protocol without using the source discovery protocol Multicast Source Discovery Protocol (MSDP) for both IPv4 and IPv6 address families.

This feature is useful when:

- Need support for IPv6 address family, which MSDP does not support.
- Interdomain connection is not required.
- Do not wish to deploy MSDP protocol.

The feature requires a Premier or Core license. For more information about licensing, see the Switch Engine v33.6.1 Licensing Guide .

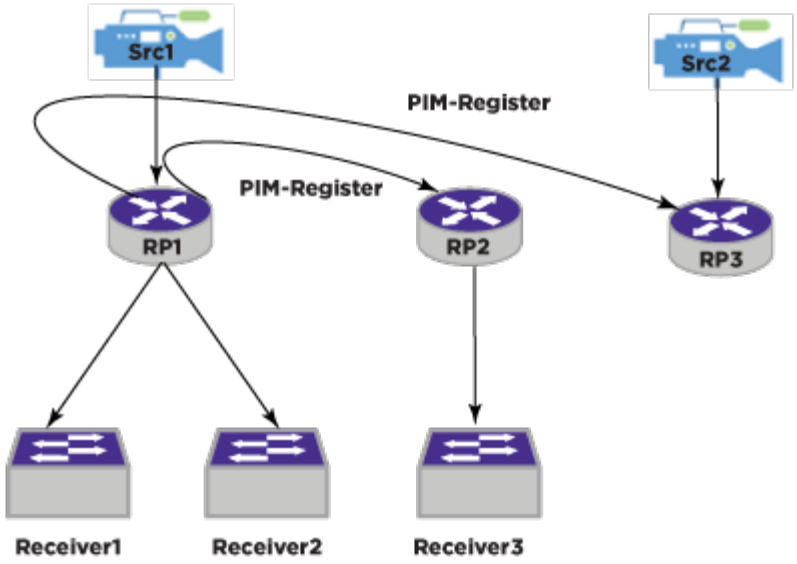


Figure 144: Anycast RP Operation

Anycast RP is a mechanism that provides fast convergence when a PIM RP node fails. Receivers and sources connect to the closest RP. The packets from a source need to get to all RPs to find joined receivers.

A unicast address is selected as the RP address. Multicast groups supported by the RP are configured using a policy file. This RP is configured either statically, or is dynamically distributed to the PIM domain using BSR protocol. To achieve the load balancing between the RPs, the same RP address is configured to multiple devices in a PIM domain. Sources and receivers connect to the nearest router selected by the L3 unicast routing protocols.

In the above topology, three RP nodes are configured with the same RP address. Recevier1 and Receiver2 choose RP1 as the nearest router. Recevier3 connects to RP2. When Source1 starts transmitting, Src1 node start sending the register requests to RP1. The same sources should be known to other receivers, such as Receiver3. Typically, in IPv4, MSDP protocol is used to propagate the learned sources from RP1 to other RPs (RP2 and RP3) through MSDP messages.

In the above topology, each node configured with an Anycast RP address and a unique unicast IP address can communicate with other Anycast RP peers configured using the policy file (see Anycast RP Using PIM Policy File Information. Anycast RP addresses and the group of nodes' unique address bindings are called an Anycast RP set. Each RP communicates with other nodes in the same RP set.

Anycast RP Using PIM Policy File Information

The policy file contains a list of Anycast RP peer addresses, and should be in the following format. This same policy must be attached to the all the RP Anycast RP peers participating in the Anycast RP set.

Sample Policy File

```
entry policy1 {
if match any{
nlri 10.10.10.1/32;
nlri 20.20.20.1/32;
nlri 30.30.30.1/32;
}
then {
permit;
}
}
```

Supported Platforms

This feature is available on platforms that support the appropriate license for the PIM feature. For complete information about software licensing, including how to obtain and upgrade your license and which licenses support the PIM feature, see the Switch Engine v33.6.1 Licensing Guide .

Limitations

The following limitation apply to this feature:

- Configuring MSDP peers between the Anycast RPs might not work as expected. Anycast RP Using PIM removes the dependence on using MSDP by removing MSDP peering between the Anycast-RPs.
- Register policy on an RP will reject the register source or groups mentioned in the policy file. If Anycast RP is configured on the same node, register policy takes precedence over the Anycast RP. Register policy filter is applied on all the PIM-Register messages received from either first hop router (FHR) or other Anycast RP peers.
- If RP is configured on an MLAG node, the peer node must not be configured as Anycast RP peer.

For information about:

- Configuring Anycast RP, see Configuring Anycast RP Using PIM (RFC 46010) on page 1192.
- Viewing Anycast RP information, see Displaying Anycast Rendezvous Point (RP) Using PIM (RFC 4610) on page 1212.

PIM Multicast Routing Over GRE Tunnels

Version 32.3 adds the ability to configure IPv4 Protocol Independent Multicast (PIM) traffic over GRE tunnels. PIM control packets are exchanged between GRE tunnel endpoints, forming a PIM neighborship across the GRE tunnel. Similar to IPv4 unicast support on GRE tunnels, multicast data packets are routed from a tenant interface to the far end of the IPv4 GRE tunnel. The IPv4 next-hop for the multicast packets is the IPv4 tunnel IP of the BGP neighbor.

Supported Platforms

5520, 5720, 7520, and 7720 series switches

Limitations

This feature has the following limitations:

- The maximum number of GRE tunnels supported is 64.
- The maximum number of VRs supported is 32.
- Routing across VRs is not supported.
- IPv6 multicast routing over tunnel is not supported.
- MSDP is not supported with this feature.
- Stacking is not suppported with this feature.

BFD Support for PIM Across All Virtual Routers

Version 33.4.1 adds support for Bidirectional Forwarding Detection (BFD) with Protocol Independent Multicast (PIM) across all Virtual Routers (VRs), including User VRs. This enhancement improves fault detection and network resilience in multicast routing environments.

PIM MLAG Enhancements

Version 33.5.1 introduces expanded support for Protocol Independent Multicast (PIM) in MLAG topologies, addressing previous limitations and improving multicast routing behavior across both default and user-created VRs.

Enhancements include:

- **PIM JOIN Forwarding over MLAG Links:** PIM JOIN messages received on MLAG ports are now checkpointed and synchronized between MLAG peers, ensuring consistent receiver state and enabling correct traffic forwarding.
- **IGMP JOIN Forwarding over MLAG Links:** IGMP JOINs continue to be supported over MLAG links, with receiver information synchronized using IGMP Sync.
- **PIM MLAG Transit Support:** Synchronization of multicast receiver state learned using PIM JOINs is now supported across MLAG transit peers. This resolves forwarding inconsistencies where only one peer previously learned receiver state. The enhancement applies to both default and user-defined VRs and leverages existing PIM checkpointing mechanisms with new message types. No CLI configuration is required to activate the feature.

These enhancements improve multicast resiliency and consistency in MLAG deployments and are automatically applied on supported platforms.

Limitations

- IPv4 PIM-DM is not supported in transit mode.
- IPv6 PIM features and PIM snooping are not supported in MLAG configurations.
- W-MLAG topologies are not supported.

IGMP Overview

IGMP is a protocol used by an IP host to register its IP multicast group membership with a router. A host that intends to receive multicast packets destined for a particular multicast address registers as a member of that multicast address group. Periodically, the router queries the multicast group to see if the group is still in use. If the group is still active, a single IP host responds to the query, and group registration is maintained.

IGMPv2 is enabled by default on the switch, and the ExtremeXOS software supports IGMPv3. However, the switch can be configured to disable the generation of periodic IGMP query packets. IGMP should be enabled when the switch is configured to perform IP multicast routing.

IETF standards require that a router accept and process IGMPv2 and IGMPv3 packets only when the router-alert option is set in received IGMP packets.

By default, the ExtremeXOS software receives and processes all IGMP packets, regardless of the setting of the router-alert option within a packet. When the switch will be used with third-party switches that expect IETF compliant behavior, use the following command to manage this feature:

```
configure igmp router-alert receive-required [on | off] {{vlan}
vlan_name}
```

```
configure igmp router-alert transmit [on | off] {{vlan} vlan_name}
```

By default, IGMP report/leave message for the local multicast address (224.0.0.x/24 groups) will always have the router-alert option set, regardless of IGMP router-alert transmit option (on and off) setting by the user.

IGMPv3, specified in RFC 3376, adds support for source filtering. Source filtering is the ability for a system to report interest in receiving packets only from specific source addresses (filter mode include) or from all sources except for specific addresses (filter mode exclude). IGMPv3 is designed to be interoperable with IGMPv1 and IGMPv2.

Note
The ExtremeXOS software supports IGMPv3 source include mode filtering, but it does not support IGMPv3 specific source exclude mode filtering.

IGMP Snooping

IGMP snooping is a Layer 2 function of the switch; it does not require multicast routing to be enabled. In IGMP snooping, the Layer 2 switch keeps track of IGMP reports and only forwards multicast traffic to that part of the local network that requires it. IGMP snooping optimizes the use of network bandwidth and prevents multicast traffic from being flooded to parts of the local network that do not need it. The switch does not reduce any IP multicast traffic in the local multicast domain (224.0.0.x).

IGMP snooping is enabled by default on all *VLAN*s and VMANs in the switch. If IGMP snooping is disabled on a VLAN or VMAN, all IGMP and IP multicast traffic floods within

the VLAN or VMAN. IGMP snooping expects at least one device on every VLAN to periodically generate IGMP query messages.

To enable or disable IGMP snooping, use the following command:

```
enable igmp snooping {forward-mcrouter-only | {vlan} name | with-proxy
vr vrname}
```

```
disable igmp snooping {forward-mcrouter-only | {vlan} name | with-proxy
vr vrname}
```

Note
IGMP snooping is not supported on SVLANs on any platform.

The IGMP snooping proxy feature represented by "with-proxy" in the above commands is enabled by default. This feature optimizes the forwarding of IGMPv1 and IGMPv2 reports. The following is true for each group:

- Only the first received IGMP join is forwarded upstream.
- Only the IGMP leave for last host is forwarded upstream.

To allow the switch to send an IGMP general query on all interfaces of a VLAN, use the following commands:

- `enable igmp proxy-query {vlan name}`
- `disable igmp proxy-query {vlan name}`

These commands are useful when either the querier is unknown or not present on a particular VLAN. This feature is enabled by default.

When a switch receives an IGMP leave message on a port, it sends a group-specific query on that port if proxy is enabled (even if it is a non-querier). The switch removes the port from the group after leave timeout (The timeout value is configurable, with a default value of 1000 ms., and a range from 0 to 175000 ms). If all the ports are removed from the group, the group is deleted and the IGMP leave is forwarded upstream. If IGMP snooping proxy is disabled, then all the IGMP reports are forwarded upstream.

Note
IGMP snooping proxy does not apply to IGMPv3 reports.

IGMP snooping is implemented primarily through ACLs, which are processed on the interfaces. These special purpose ACLs are called IGMP snooping hardware filters. The software allows you to choose between two types of IGMP snooping hardware filters: per-port filters (the default) and per-VLAN filters.

The two types of IGMP snooping hardware filters use switch hardware resources in different ways. The two primary hardware resources to consider when selecting the IGMP snooping hardware filters are the Layer 3 multicast forwarding table, and the interface *ACL (Access Control List)*s. The size of both of these hardware resources is determined by the switch model. In general, the per-port filters consume more

resources from the multicast table and less resources from the ACL table. The per-VLAN filters consume less space from the multicast table, and more from the ACL table.

Using the per-port filters can fill up the multicast table and place an extra load on the CPU. To avoid this, configure the switch to use the per-VLAN filters.

Note
The impact of the per-VLAN filters on the ACL table increases with the number of VLANs configured on the switch. If you have a large number of configured VLANs, we suggest that you use the per-port filters.

IGMP Snooping Filters

IGMP snooping filters allow you to configure a policy file on a port to allow or deny IGMP report and leave packets coming into the port. (For details on creating policy files, see Policy Manager on page 1448.) The IGMP snooping filter feature is supported by IGMPv2 and IGMPv3.

Note
Do not confuse IGMP snooping filters with IGMP snooping hardware filters explained in previous section. IGMP snooping filters are software filters, and the action is applied at the software level by the ExtremeXOS multicast manager.

For the policies used as IGMP snooping filters, all the entries should be IP address type entries, and the IP address of each entry must be in the class-D multicast address space but should not be in the multicast control subnet range (224.0.0.x/24).

1. Use the following template to create a snooping filter policy file that denies IGMP report and leave packets for the 239.11.0.0/16 and 239.10.10.4/32 multicast groups:

```
#
# Add your group addresses between "Start" and "end"
# Do not touch the rest of the file!!!!
entry igmpFilter {
if match any {
#------------------ Start of group addresses ------------------
nlri  239.11.0.0/16;
nlri  239.10.10.4/32;
#------------------- end of group addresses -------------------
} then {
deny;
}
}
entry catch_all {
if {
} then {
permit;
}
}
```

2. After you create a policy file, use the following command to associate the policy file and filter to a set of ports:

   `configure igmp snooping vlan` *`vlanname`* **`ports`** *`portlist`* **`filter`** [*`policy`* | **`none`**]

3. To remove the filter, use the **`none`** option.

4. To display the IGMP snooping filters, use the following command:

   ```
   show igmp snooping {vlan} name filter
   ```

Static IGMP

To receive multicast traffic, a host must explicitly join a multicast group by sending an *IGMP* report; then, the traffic is forwarded to that host. In some situations, you might like multicast traffic to be forwarded to a port where a multicast-enabled host is not available (for example, when you test multicast configurations).

Static IGMP emulates a host or router attached to a switch port, so that multicast traffic is forwarded to that port, and the switch sends a proxy join for all the statically configured IGMP groups when an IGMP query is received. You can emulate a host to forward a particular multicast group to a port; and you may emulate a router to forward all multicast groups to a port. Static IGMP is only available with IGMPv2.

- To emulate a host on a port, use the following command:

  ```
  configure igmp snooping {vlan} vlan_name {ports port_list}add static group ip_address
  ```

- To emulate a multicast router on a port, use the following command:

  ```
  configure igmp snooping {vlan} vlan_name {ports port_list}add static router
  ```

- To remove these entries, use the corresponding commands:

  ```
  configure igmp snooping {vlan} vlan_name {ports port_list }delete static group [ip_address | all]
  ```

  ```
  configure igmp snooping vlan vlan_name {ports port_list } delete static router
  ```

- To display the IGMP snooping static groups, use the following command:

  ```
  show igmp snooping {vlan} vlan_name static [group | router]
  ```

IGMP Loopback

Prior to ExtremeXOS 15.3.2, you could configure static groups, but it was necessary to specify port(s). As of ExtremeXOS Release 15.3.2, the configuration of dynamic groups is supported. The *IGMP* Loopback feature, along with the existing static group feature, supports the configuration of static and/or dynamic groups with or without ports.

A *VLAN* in loopback mode may not have ports associated with it, but its operational status is up. However, it is not possible to have multicast receivers on a VLAN without having a port. Sometimes there is a need to pull the multicast traffic from upstream to the loopback VLAN for troubleshooting. The traffic need not always be forwarded to any ports/receivers. The IGMP Loopback feature allows you to configure groups on a VLAN without specifying a port, so the traffic is pulled from upstream but not forwarded to any port.

The loopback (Lpbk) port is the logical port on a VLAN in the application context. If you configure a group on a VLAN but do not specify the port, the switch forms an IGMPv2 join and assumes it to be received on the Lpbk port. A dynamic group ages out after

the membership timeout if there are no other receivers. Membership joins refresh the dynamic group. The static group remains until it is removed from the configuration.

The following command is used to configure static or dynamic group entry :

```
configure igmp snooping {vlan} vlan_name {ports port_list} add [static | dynamic] group ip_address
```

Limiting the Number of Multicast Sessions on a Port

The default configuration places no limit on the number of multicast sessions on each *VLAN* port. To place a limit on the number of learned MLD reports, or remove the limit, use the command:

```
configure mld snooping {vlan} vlan_name ports port_list join-limit [num_joins | no-limit]
```

Enabling and Disabling IGMP Snooping Fast Leave

When the fast leave feature is enabled and the last host leaves a multicast group, the router immediately removes the port from the multicast group. The router does not query the port for other members of the multicast group before removing the group, and/or the port.

The default setting for *IGMP* snooping fast leave is disabled.

- To enable the fast leave feature, use the following command:

  ```
  enable igmp snooping {vlan} name fast-leave
  ```

- To disable the fast leave feature, use the following command:

  ```
  disable igmp snooping {vlan} name fast-leave
  ```

IGMP-SSM Mapping

The *IGMP*-SSM Mapping feature allows IGMPv1 and IGMPv2 hosts to participate in SSM functionality, and eliminates the need for IGMPv3. You can configure SSM map entries that specify the sources used for a group/group range for which SSM functionality has to be applied. You also have the option to configure the domain name and DNS server to use to obtain the source list.

When a IGMPv1 or IGMPv2 report is received, the configured sources are provided to PIM so that it can send source specific joins. When the host leaves or when the membership times out, PIM is informed so that it can consider sending prunes.

In a multi-access network (where more than one router is receiving IGMP messages from the hosts), only the designated router sends joins towards the source, so it is desirable to have same configuration for SSM group range and SSM Mapping range on all routers in a *VLAN*.

Limitations

The following limitations apply:

- A single group address or range can be mapped to a maximum of 50 sources. If more than 50 sources are configured, the switch uses the 50 longest-matching prefixes.
- We recommend a maximum of 500 mappings per switch, but no limit is imposed by the software.

Limitations

The following features of Change-of-Authorization (RFC5176) are not implemented in ExtremeXOS:

- Reverse Path Forwarding Check—Typically this is used in a proxy scenario. This check is used to determine if the IP address indicated by the *RADIUS (Remote Authentication Dial In User Service)* attributes is a routable destination address for a request sent by the switch software.
- IPSEC encryption—End-to-end encryption of both the RADIUS requests and responses.
- Disconnect-Request and Change-of-Authorization packets identifying sessions with anything other than the Calling-Station-Id attribute containing a properly formatted MAC address. In addition to the Calling-Station-ID attribute, you can also use a NAS-Port attribute, which indicates the index of the specific port the session is connected to.
- Acct-Session-Id attribute—This is an alternate means of session identification. Sessions are currently uniquely identified by port and MAC address pair.
- Retransmissions of Disconnect-Request or Change-of-Authorization ACK and NAK packets—Retransmissions of packets is the responsibility of the device initiating the dynamic authorization transactions.

Configuring IGMP-SSM Mapping

To support PIM-SSM for IGMPv1 and IGMPv2 clients, a PIM-SSM range must be configured, and that range should include all groups to which the clients want access. If IGMPv1 and IGMPv2 clients request group addresses outside the PIM-SSM range, those addresses are ignored by PIM-SSM and forwarded to PIM as (*, G) requests.

- To enable *IGMP*-SSM mapping, first configure a PIM-SSM range, and then enable IGMP-SSM mapping using the following commands:

  ```
  configure pim ssm range [default | policy policy-name]
  enable igmp ssm-map {vr vr-name}
  ```

- To configure an IGMP-SSM mapping, use the following command:

  ```
  configure igmp ssm-map add group_ip [prefix | mask] [source_ip |
  src_domain_name] {vr vr-name}
  ```

- To remove a single IGMP-SSM mapping, use the following command:

  ```
  configure igmp ssm-map delete group_ip [ prefix} | mask] [source_ip |
  all] vr vr-name }
  ```

- To remove all IGMP-SSM mappings on a , use the following command:
  `unconfigure igmp ssm-map {vr vr-name}`
- To disable IGMP-SSM mapping on a virtual router, use the following command:
  `disable igmp ssm-map {vr vr-name}`

Displaying IGMP-SSM Mappings

To see whether or not *IGMP*-SSM mapping is enabled or disabled and to view the configured mappings for a multicast IP address, use the following command:

`show igmp ssm-map {group_ip} {vr vr-name}`

IGMP Statistics

To view IGMP counters, use the following command:

`show igmp counters`

To clear IGMP counters, use the following command:

`clear igmp counters`

Configuring EAPS Support for Multicast Traffic

The ExtremeXOS software provides several commands for configuring how *EAPS (Extreme Automatic Protection Switching)* supports multicast traffic after an EAPS topology change. For more information, see the descriptions for the following commands:

`configure eaps multicast add-ring-ports [on | off]`

`configure eaps multicast send-query [on | off]`

`configure eaps multicast temporary-flooding [on | off]`

These commands apply for both *IGMP* and MLD.

Note
Using the `configure eaps multicast send-query` command applies to both IGMP and MLD. This also replaces the `configure eaps multicast send-igmp-query` command.

Configuring IP Multicast Routing

Enabling Multicast Forwarding

To enable IP multicast forwarding:

1. Configure the system for IP unicast routing.
2. Enable multicast forwarding on the interface.
   `enable ipmcforwarding {vlan name}`

Configuring PIM

To configure PIM multicast routing, enable multicast forwarding as described in Enabling Multicast Forwarding on page 1190 and do the following:

1. Configure PIM on all IP multicast routing interfaces using the following command:

   `configure pim {ipv4 | ipv6} add vlan [vlan-name | all] {dense | sparse} {passive}`

2. To enable and configure the PIM-DM state refresh feature on one or all VLANs, use the following commands:

   `configure pim {ipv4 | ipv6} state-refresh {vlan} [vlan_name | all] [on | off]`

   `configure pim { ipv4 | ipv6 } state-refresh timer origination-interval interval`

   `configure pim {ipv4 | ipv6} state-refresh timer source-active-timer interval`

   `configure pim {ipv4 | ipv6} state-refresh ttl ttlvalue`

3. For PIM-SSM, specify the PIM-SSM range, enable IGMPv3, and enable PIM-SSM on the interfaces using the following commands:

   `configure pim {ipv4 | ipv6} ssm range [default | policy policy-name]`

   `enable igmp {vlan vlan name } {IGMPv1 | IGMPv2 | IGMPv3}`

   `enable pim {ipv4 | ipv6} ssm vlan [vlan_name | all]`

4. Enable PIM on the router using the following command:

   `enable pim`

Configuring Multicast Static Routes

Note
Multicast static routes are supported in the IPv4 address family, but not the IPv6 address family.

Static routes are used to reach networks not advertised by routers, and are manually entered into the routing table.

- You can use either of two commands to create multicast static routes. The recommended command is the following:

  `configure iproute add [ipNetmask | ip_addr mask] gateway {bfd} {metric} {multicast | multicast-only | unicast | unicast-only} {vr vrname}`

  For example:

  ```
  configure iproute add 55.1.10.0/24 44.1.12.33 multicast
  ```

- The following command is still supported for backward compatibility with earlier ExtremeXOS software releases:

  `configure ipmroute add [source-net mask-len | source-net mask] {{protocol} protocol} rpf-address {metric} {vr vr-name}`

In the following example, the configure ipmroute add command is used to specify protocol information for a route:

```
configure ipmroute add 58.1.10.0/24 ospf 44.1.12.33
```

Note
Ipmroute not supported for IPv6.

When a static route is configured with protocol information, the route is shown as UP only when the protocol route is available. Otherwise, the route is Down. In the example above, the multicast static route 58.1.10.0/24 is shown as UP only when the *OSPF* route is available to reach the network 58.1.10.0/24.

Static routes are stored in the switch configuration and can be viewed with the show configuration command. Static multicast routes that do not include protocol information are displayed using the configure iproute command format, even if they were created using the configure ipmroute command. Static routes that are created with a protocol field are displayed using the configure ipmroute command format.

Disabling IP Multicast Compression

The IP multicast compression feature is enabled by default. You should only disable this feature if you suspect that switch processing resources are limited or if you think this feature is causing problems on the switch.

- To disable or enable IP multicast compression, use the following command:

  ```
  configure forwarding ipmc compression {group-table | off}
  ```

Configuring Anycast RP Using PIM (RFC 46010)

To configure or remove Anycast RPs, use the following command:

```
configure pim {ipv4 | ipv6} anycast-rp ip_address [policy | none]
```

Multicast Over MLAG Configuration

For sample topology for , see Layer 3 IP Multicast using PIM-SM.

PIM Configuration Examples

PIM-DM Configuration Example

In Figure 145, the system labeled IR 1 is configured for IP multicast routing, using PIM-DM.

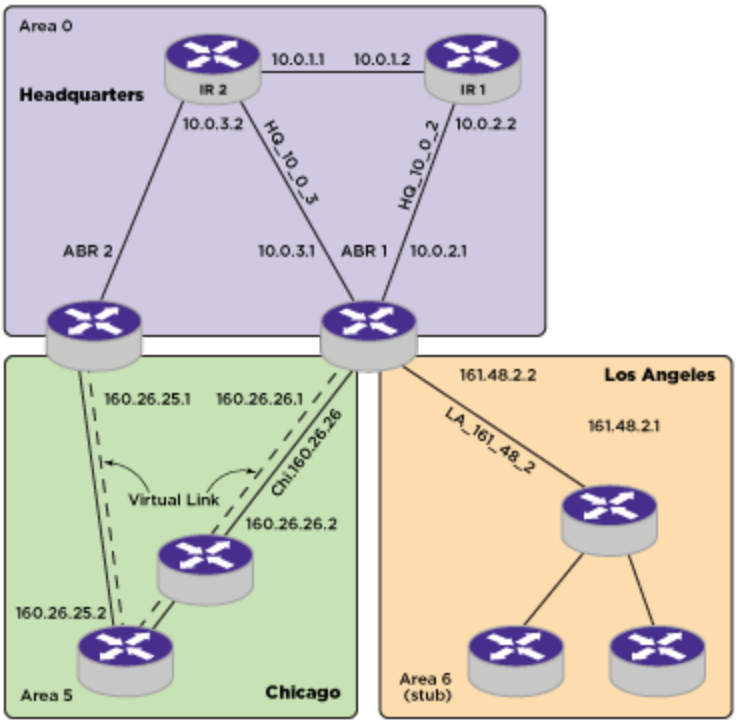


Figure 145: IP Multicast Routing Using PIM-DM Configuration Example

Note
The above figure is used in the the OSPF section to describe the *OSPF* configuration on a switch.

The router labeled IR1 has the following configuration:

```
configure vlan HQ_10_0_1 ipaddress 10.0.1.2 255.255.255.0
configure vlan HQ_10_0_2 ipaddress 10.0.2.2 255.255.255.0
configure ospf add vlan all area 0.0.0.0
enable ipforwarding
enable ospf
enable ipmcforwarding
configure pim add vlan all dense
enable pim
configure pim state-refresh vlan all on
```

PIM-SM Configuration Example

In Figure 146, the system labeled ABR1 is configured for IP multicast routing using PIM-SM.

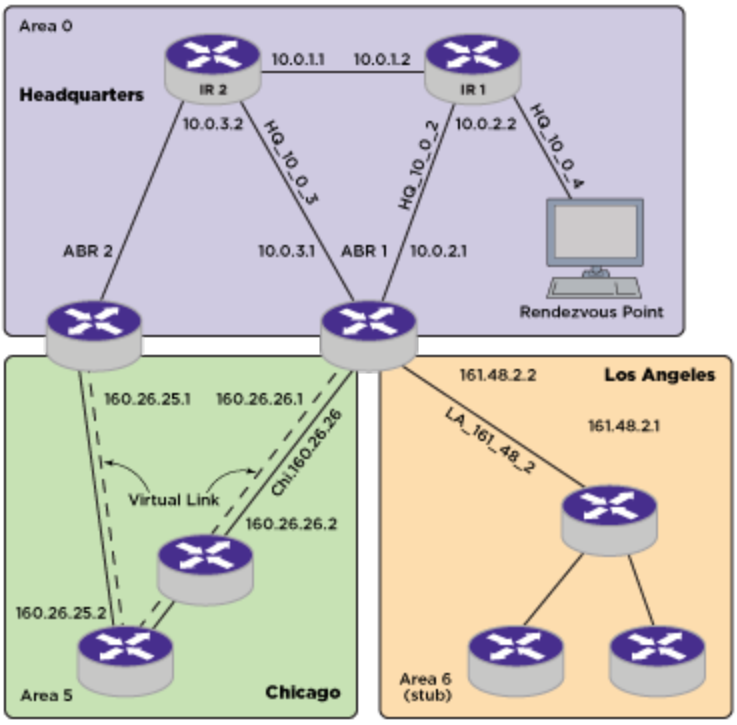


Figure 146: IP Multicast Routing Using PIM-SM Configuration Example

Note
The above figure is used in the OSPF section to describe the *OSPF* configuration on a switch.

The router labeled ABR1 has the following configuration:

```
configure vlan HQ_10_0_2 ipaddress 10.0.2.1 255.255.255.0
configure vlan HQ_10_0_3 ipaddress 10.0.3.1 255.255.255.0
configure vlan LA_161_48_2 ipaddress 161.48.2.2 255.255.255.0
configure vlan CHI_160_26_26 ipaddress 160.26.26.1 255.255.255.0
configure ospf add vlan all area 0.0.0.0
enable ipforwarding
enable ipmcforwarding
configure pim add vlan all sparse
tftp TFTP_SERV -g -r rp_list.pol
configure pim crp HQ_10_0_3 rp_list 30
configure pim cbsr HQ_10_0_3 30
```

The policy file, rp_list.pol, contains the list of multicast group addresses serviced by this RP. This set of group addresses are advertised as candidate RPs. Each router then elects the common RP for a group address based on a common algorithm. This group to RP mapping should be consistent on all routers.

The following is a policy file that configures the CRP for the address ranges 239.0.0.0/24 and 232.144.27.0:

```
entry extreme1 {
    if match any {
    }
        then {
            nlri  239.0.0.0/24 ;
```

```
            nlri  232.144.27.0/24 ;
    }
}
```

PIM-SSM Configuration Example

In the following example, the default PIM-SSM range of 232.0.0.0/8 is configured. For all interfaces, non-IGMPv3 messages and IGMPv3 exclude messages are ignored for addresses in this range. Hosts that use IGMPv3 on *VLAN* v13 can request and receive source specific multicast streams for addresses in the PIM-SSM range.

```
create vlan v12
create vlan v13
configure v12 add port 1
configure v13 add port 2
configure v12 ipaddress 12.1.1.1/24
configure v13 ipaddress 11.1.1.1/24
configure pim add vlan all sparse
enable ipforwarding
enable ipmcforwarding
enable igmp IGMPv3
configure pim ssm range default
enable pim ssm vlan v13
enable pim
```

Note
(*;G)s are created for groups outside the SSM range. SSM may not be enabled for the ingress VLAN (see Table 79 on page 1168).

PIM Snooping Configuration Example

Figure 147 shows a network configuration that supports PIM snooping.

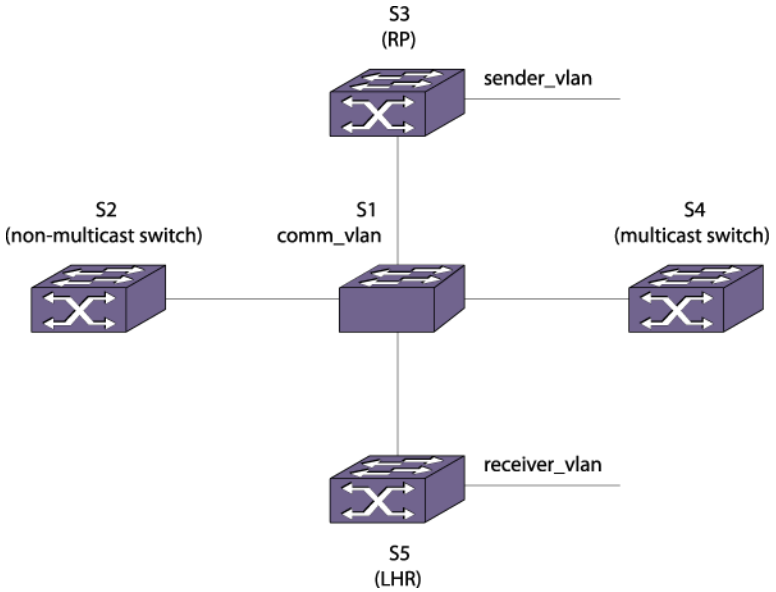


Figure 147: PIM Snooping Configuration Example

In Figure 147 Layer 3 switches S2, S3, S4, and S5 are connected using the Layer 2 switch S1 through the *VLAN* comm_vlan. Switches S3, S4, and S5 are multicast capable switches, and switch S2 is a non-multicast capable switch, which has no multicast routing protocol running.

Without PIM snooping, any ingress multicast data traffic on comm_vlan is flooded to all the switches, including switch S2, which does not support multicast traffic. *IGMP* snooping does not reduce flooding because it floods the multicast traffic to all router ports.

The network design calls for most multicast traffic to egress switch S5. PIM snooping helps by intercepting the PIM joins received from the downstream routers and forwarding multicast traffic only to those ports that received PIM joins.

Switch S1 (PIM Snooping Switch) Configuration Commands

The following is an example configuration for the PIM snooping switch S1:

```
create vlan comm_vlan
configure vlan comm_vlan add port 1,2,3,4
disable igmp snooping
disable igmp_snooping comm_vlan
enable pim snooping
enable pim snooping comm_vlan
```

Switch S3 Configuration Commands

The following is an example configuration for switch S3, which also serves as an RP:

```
create vlan comm_vlan
configure vlan comm_vlan add port 1
configure comm_vlan ipa 10.172.168.4/24
enable ipforwarding comm_vlan
enable ipmcforwarding comm._vlan
configure pim add vlan comm_vlan sparse
configure ospf add vlan comm._vlan area 0.0.0.0

create vlan sender_vlan
configure vlan sender_vlan add port 2
configure sender_vlan ipa 10.172.169.4/24
enable ipforwarding comm_vlan
enable ipmcforwarding comm._vlan
configure pim add vlan comm._vlan sparse
configure ospf add vlan comm_vlan area 0.0.0.0

enable pim
enable ospf

configure pim crp static 10.172.169.4 pim_policy // RP is configured using the policy
pim_policy for the group 224.0.0.0/4
```

Switch S5 Configuration Commands

The following is an example configuration for switch S5, which serves as the last hop router for multicast traffic:

```
create vlan comm_vlan
configure vlan comm_vlan add port 1
configure comm_vlan ipa 10.172.168.2/24
enable ipforwarding comm_vlan
enable ipmcforwarding comm._vlan
configure pim add vlan comm_vlan sparse
configure ospf add vlan comm._vlan area 0.0.0.0

create vlan receiver_vlan
configure vlan sender_vlan add port 1
configure sender_vlan ipa 10.172.170.4/24
```

```
enable ipforwarding comm_vlan
enable ipmcforwarding comm._vlan
configure pim add vlan comm._vlan sparse
configure ospf add vlan comm_vlan area 0.0.0.0

enable pim
enable ospf

configure pim crp static 10.172.169.4 pim_policy // RP is configured using the policy
pim_policy for the group 224.0.0.0/4
```

Switch S4 Configuration Commands

The following is an example configuration for switch S4, which is neither an LHR nor a RP:

```
create vlan comm_vlan
configure vlan comm_vlan add port 1
configure comm_vlan ipa 10.172.168.3/24
enable ipforwarding comm_vlan
enable ipmcforwarding comm._vlan
configure pim add vlan comm_vlan sparse
configure ospf add vlan comm._vlan area 0.0.0.0

enable pim
enable ospf

configure pim crp static 10.172.169.4 pim_policy // RP is configured using the policy
pim_policy for the group 224.0.0.0/4
```

Switch S2 Configuration Commands

The following is an example configuration for switch S2, which is not enabled for PIM:

```
create vlan comm_vlan
configure vlan comm_vlan add port 1
configure comm_vlan ipa 10.172.168.6/24
enable ipforwarding comm_vlan
enable ipmcforwarding comm._vlan
configure ospf add vlan comm._vlan area 0.0.0.0

enable ospf
```

PIM Snooping Example Configuration Displays

After the example configuration is complete, multicast receivers connect to the network through switch S5 and multicast sources connect through switch S3.

When switch S5 receives an IGMP request from the receiver_vlan for group 225.1.1.1, it sends a PIM (*, G) join towards switch S3, which is the RP. The PIM snooping feature on switch S1 snoops the (*, G) join, and the resulting entry can be viewed by entering the following command at switch S1:

```
# show pim snooping vlan comm_vlan
PIM Snooping                          ENABLED
Vlan comm_vlan(3971)               Snooping ENABLED
Source          Group           RP               UpPort   DownPort   Age      HoldTime
*               225.1.1.1       10.172.169.4        1        2          15        210
Neighbor IP      DR     Port       Age          Hold Time
10.1272.168.4    YES     1          2            105
10.1272.168.2    NO      2          2            105
10.1272.168.3    NO      3          2            105
```

Once multicast traffic arrives from the sender_vlan, the LHR (switch S2) sends the (S, G) join message, which is snooped by the PIM snooping switch, switch S1. The resulting entries can be viewed by entering the following command at switch S1:

```
# show pim snooping vlan comm_vlan
PIM Snooping                              ENABLED
Vlan comm_vlan(3971)                   Snooping ENABLED
Source          Group           RP               UpPort   DownPort   Age      HoldTime
*               225.1.1.1       10.172.169.4        1        2          15        210
10.172.169.10   225.1.1.1       10.172.169.4        1        2          15        210
Neighbor IP      DR     Port       Age          Hold Time
10.1272.168.4    YES     1          2             105
10.1272.168.2    NO      2          2             105
10.1272.168.3    NO      3          2             105
```

Multicast traffic is forwarded only to those ports that have received (*, G) or (S, G) joins and designated router (DR) ports.

Configuring PIM Multicast Routing Over GRE Tunnels

The following commands support this feature:

To configure a PIM tunnel, use `configure pim {ipv4} add {tunnel} [tunnel_name | tunnel all] {sparse | dense} {passive}`

To delete a PIM tunnel, use `configure pim {ipv4} delete tunnel [tunnel_name | tunnel all]`

To configure a PIM timer, use `configure pim {ipv4} timer hello_interval jp_interval [{tunnel} tunnel_name | tunnel all]`

To configure a PIM dr-priority, use `configure pim {ipv4} [{tunnel} tunnel_name | tunnel all] dr-priority priority`

To disable IPMC forwarding tunnels, use `disable ipmcforwarding {tunnel tunnel_name}`

To disable a PIM SSM, use `disable pim {ipv4} ssm tunnel [tunnel_name | tunnel all]`

To enable IPMC forwarding tunnels, use `enable ipmcforwarding {tunnel tunnel_name}`

To enable a PIM SSM tunnel, use `enable pim {ipv4} ssm tunnel [tunnel_name | tunnel all]`

To display PIM tunnel information, use `show pim tunnel`

To unconfigure a PIM tunnel, use `unconfigure pim {ipv4}{tunnel tunnel_name}`

Multicast VLAN Registration

Multicast *VLAN* Registration (MVR) is designed to support distributing multicast streams for IPTV to subscribers over a Layer 2 network. In a standard Layer 2 network, a multicast stream received on a VLAN is not forwarded to another VLAN. The streams are confined to the Layer 2 broadcast domain. In an *IGMP* snooping environment, streams are forwarded only to interested hosts on a VLAN. For inter-VLAN forwarding (routing) a multicast routing protocol, such as PIM/DVMRP must be deployed.

MVR breaks this basic rule, so that a stream received over Layer 2 VLANs is forwarded to another VLAN, eliminating the need for a Layer 3 routing protocol. It simplifies the multicast stream distribution and is a better solution for IPTV-like services. With MVR, a multicast stream is forwarded to all VLANs containing interested hosts.

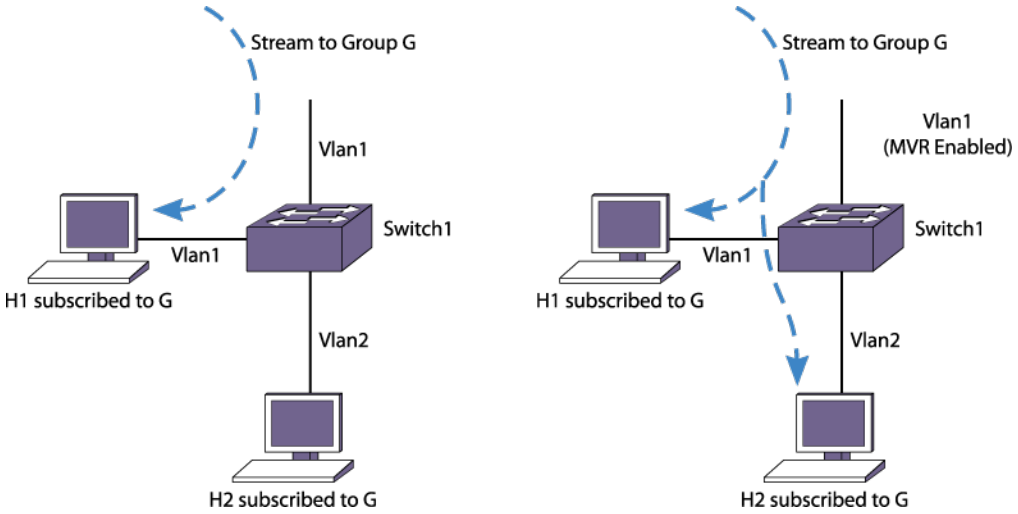


Figure 148: Standard VLAN Compared to an MVR VLAN

In the above figure, the left side shows a standard VLAN carrying a multicast stream.

The host H1 receives the multicast stream because it resides on VLAN Vlan1, but host H2 does not receive the multicast traffic because no IP multicast routing protocol forwards the stream to VLAN Vlan2. On the right side of the figure, H2 does receive the multicast stream. Because Vlan1 was configured as an MVR VLAN, the multicast stream is forwarded to the other VLANs on the switch. containing hosts that have requested the stream. To configure a VLAN as an MVR VLAN, use the following command: `configure mvr add vlan vlan-name`

Typically, IGMP snooping is enabled, so only hosts that have requested a stream can see the multicast traffic. For example, another host on VLAN2 cannot receive the traffic unless it has sent an IGMP request to be included in the group.

Notice that only VLAN1 is MVR enabled. Configure MVR only on the ingress VLAN. To enable MVR on the switch, use the following command: `enable mvr`

Basic MVR Deployment

Because MVR is primarily targeted for IPTV and similar applications, a basic deployment for that application is shown Figure 149. In the figure, an IPTV server is

connected through a router to a network of switches. Switch 1 has three customer VLANs, Vlan2, Vlan3, and Vlan4. The multicast streams are delivered through the network core (Metro Ethernets), which often use a ring topology and some kind of redundant protection to provide high availability. For example, McastVlan forms a ring through switches Switch1 through Switch4. The link from Switch2 to Switch4 is shown as blocked, as it would be if some form of protection (such as *EAPS*) is used.

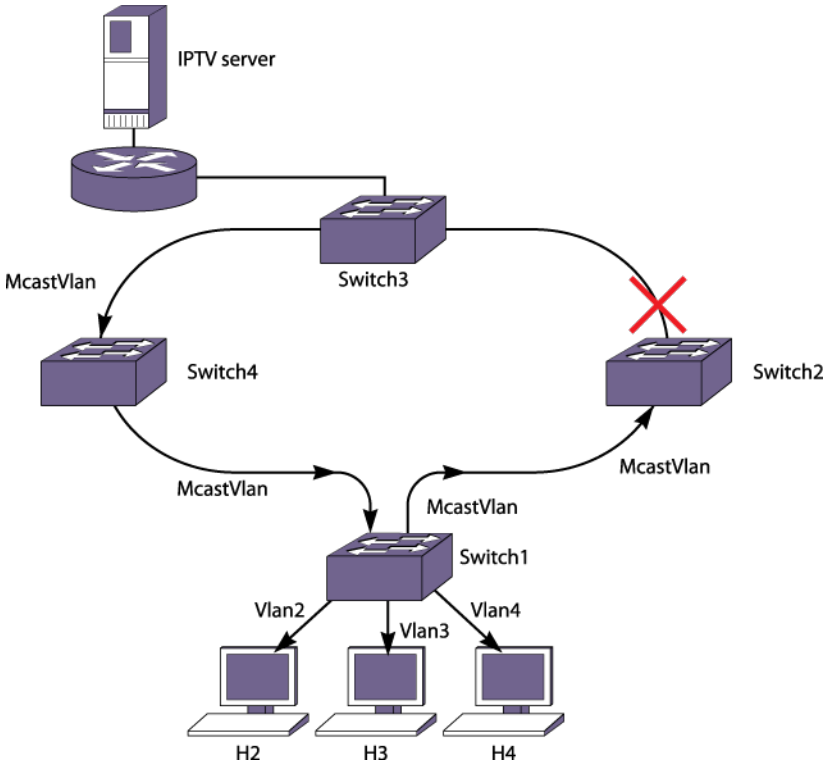


Figure 149: Basic MVR Deployment

Without MVR, there are two ways to distribute multicast streams in this topology:

- Extend subscriber *VLAN*s (Vlan2, Vlan3, and Vlan4) to the network core, by tagging the ports connecting the switches.
- Configure all VLANS with an IP address and run PIM or DVMRP on each switch.

There are problems with both of these approaches. In the first approach, multiple copies of the same stream (IPTV channel) would be transmitted in the core, wasting bandwidth. In the second approach, all switches in the network core would have to be Layer 3 multicast aware, and would have to run a multicast protocol. Typical network cores are Layer 2 only.

MVR provides a simple solution to this problem If McastVlan in Switch1 is configured with MVR, it leaks the traffic into the local subscriber VLANs that contain hosts that request the traffic. For simple cases, perform these configuration steps:

- Configure MVR on McastVlan.
- Configure an IP address and enable *IGMP* and IGMP snooping on the subscriber VLANs (by default IGMP and IGMP snooping are enabled on Extreme Networks switches).

- For all the multicast streams (IPTV channels), configure static IGMP snooping membership on the router on McastVlan.
- Enable MVR on the switches in the network.

Note
MVR works best with IGMPv1 and IGMPv2. We recommend that you do not use MVR with IGMPv3.

The strategy above conserves bandwidth in the core and does not require running PIM on the subscriber switches.

In this topology, a host (for example, a cable box or desktop PC) joins a channel through an IGMP join message. Switch1 snoops this message and adds the virtual port to the corresponding cache's egress list. This is possible because an MVR enabled VLAN can leak traffic to any other VLAN. When the user switches to another channel, the host sends an IGMP leave for the old channel and a join for the new channel. The corresponding virtual port is removed from the cache for the old channel and is added to the cache for the new channel.

As discussed in Static and Dynamic MVR on page 1201, McastVlan also proxies IGMP joins learned on other VLANs to the router. On an MVR network it is not mandatory to have a router to serve the multicast stream. All that is required is to have a designated IGMP querier on McastVlan. The IPTV server can also be directly connected to McastVlan.

Static and Dynamic MVR

Static MVR

In a typical IPTV network, there are several high demand basic channels. At any instant there is at least one viewer for each of these channels (streams), and they should always be available at the core (ring). When a user requests one of these channels, it is quickly pulled locally from the multicast *VLAN*. You have the option to use the static router configuration in each of the switches in the core. But this will cause all the channels to be available in the core, which may not be desired. For example, on an Extreme Networks router, you can use the following commands:

```
configure igmp snooping {vlan} vlan_name ports port_list add static
router
```

You can use the static MVR configuration and choose the groups for which the multicast stream should be flooded. If a multicast stream for a group in the static MVR range is received on an MVR enabled VLAN, it is always flooded on the MVR VLAN. This allows the neighbor switch in the ring to receive all the static MVR streams.

Dynamic MVR

In contrast, since a video content provider would like to provide a variety of on-demand and other premium channels, there are often many lower demand (fewer viewers) premium channels that cannot all be made available simultaneously at the core network. These should be streamed from the router only if requested by a host.

IGMP is the standard method used by a host to request a stream. However, IGMP packets are constrained to a VLAN. Thus, subscribers' IGMP join requests on the VLAN cannot be forwarded onto other VLANS. You can use a dynamic MVR configuration, and choose the groups for which the IGMP join requests should be proxied over the MVR VLAN. Thus, in Figure 1, McastVlan sends join and leave messages on behalf of Vlan2, Vlan3, and Vlan4. The router receives the messages on McastVlan and streams corresponding channels onto the core network. This provides on-demand service, and an administrator does not need to configure static IGMP on the router for each of these channels.

Configuring MVR Address Range

By default, all multicast groups belong to MVR address range. Use the following command to specify the MVR address range:

```
configure mvr vlan vlan-name mvr-address {policy-name | none}
```

- Only the groups listed in the policy with "allow" action belong to MVR address range.
- For non-MVR groups, snoop cache is created and join requests are not proxied over MVR *VLAN*.

Configuring Static and Dynamic MVR

By default, all MVR streams are static.

- Use the following command to specify which groups are static:
  ```
  configure mvr vlan vlan-name static group {policy-name | none}
  ```
  - Only the groups listed in policy with "allow" action are static. Any other groups in MVR address range are dynamic.
  - The groups in policy with "deny" action are dynamic. Any other groups in MVR address range are also dynamic, unless explicitly configured in the policy with "allow" action.
  - A group should not be configured to be both static and dynamic.

MVR Configuration Example

The following example configuration is a two DUT scenario, in L2 mode , with no routing protocol or PIM configured.

- DUT-1 is sender
- DUT-2 is receiver
- *VLAN* v1 spans over DUT-1 and DUT-2 , DUT-2 also has v2 where *IGMP* joins are coming IN (225.1.1.1)
- DUT-2 has a VLAN v3, which also has a receiver connected sending IGMP join for same group as v2 (225.1.1.1)
- VLAN v1 in the DUT-2 has another port apart from the trunk port, no joins are being sent on this port.

Configure MVR on vlan v1 on DUT-2

```
# show config "mcmgr"
#
# Module mcmgr configuration.
#
```

```
enable mvr vr VR-Default
configure mvr add vlan v1
configure mvr vlan v1 mvr-address none
configure mvr vlan v1 static group none*
 *5420F-48P-4XE.158 #
```

The traffic will be flooded for the group only on MVR vlan (v1).

Since there are IGMP joins coming in on v2 and v3, v2, v3, and the second port in the MVR vlan v1 will receive traffic.

Configure the following policy file;

```
# vi mvrPolicy.polentry policy1 {
if match any {
    nlri 225.1.1.1/24;
} then {
    permit;
}
}
-------------------------
```

When applying this policy file under static group on DUT-2

```
# configure mvr vlan v1 static group mvrPolicy
#configure mvr vlan v1 mvr-address none
```

When the policy file contains "permit", the traffic flows to v2, v3, and the second port in MVR VLAN.

When the policy file is changed to "deny", the second port in the MVR VLAN v1 will stop receiving the traffic.

If you configure static policy (by default – permit), traffic for that group range will always be available in the MVR VLAN, that is, it will be forwarded to all the ports in MVR VLAN.

When applying this policy file under mvr-address (Dynamic) on DUT-2:

```
# configure mvr vlan v1 static group none
# configure mvr vlan v1 mvr-address mvrPolicy
```

When the policy file contains "permit", the traffic flows to v2,v3 and the second port in MVR vlan.

When the policy file is changed to "deny", the second port in MVR VLAN v1 continues receiving traffic, but VLAN v2 and v3 stop receiving traffic, in spite of IGMP groups being learned. This is because the join on v2 and v3 will not be leaked to MVR VLAN.

Essentially, the dynamic policy does not directly apply on traffic, but it affects the joins, based on which traffic is forwarded or blocked.

Dynamic means only if the join is sent then traffic is forwarded.

The join is leaked to MVR VLAN so traffic from MVR VLAN will be received by other VLANs (v2 and v3).

To confirm if a join was leaked to MVR VLAN use the `show igmp group` command. It should have the group learned on MVR VLAN (v1) with port as "MVR".

MVR Forwarding

The goal for MVR is to limit the multicast traffic in the core Layer 2 network to only the designated multicast *VLAN*. If the backbone Layer 2 port is tagged with multiple VLANs, as shown in Figure 150 on page 1204, a set of rules is needed to restrict the multicast streams to only one VLAN in the core.

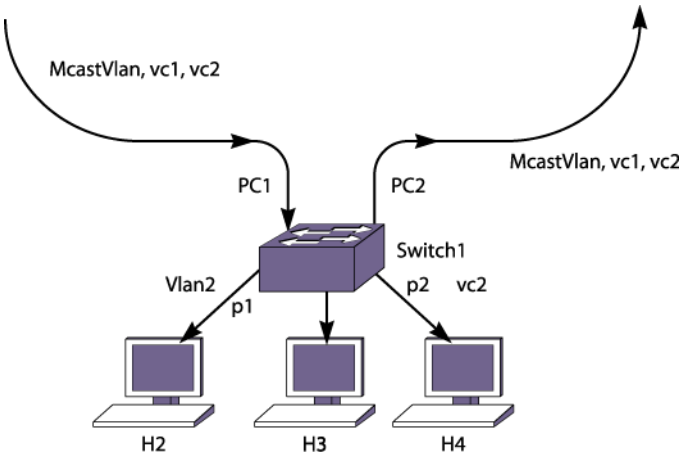


Figure 150: Multiple VLANs in the Core Network

The core network has 2 more VLANs, vc1 and vc2, to provide other services. With MVR, multicast traffic should be confined to McastVlan, and should not be forwarded to vc1 and vc2. It should be noted that MVR is configured only on the ingress VLAN (McastVlan). MVR is not configured on any other VLANs.

In the same way as the *IGMP* snooping forwarding rules, the multicast stream is forwarded onto member ports and router ports on the VLAN. For a stream received on MVR enabled ports, this rule is extended to extend membership and router ports to all other VLANs. This rule works well on the topology in Figure 150. However, in a tagged core topology, this rule forwards traffic onto VLANs, such as vc1 and vc2, on ports PC1 and PC2. This results in multiple copies of same stream on the core network, thus reintroducing the problem that MVR was intended to solve.

To avoid multiple copies of the same stream, MVR forwards traffic with some special restrictions. MVR traffic is not forwarded to another VLAN unless a host is detected on the port. On the ingress MVR VLAN, packets are not duplicated to ports belonging to MVR VLANs. This is to prevent duplicate multicast traffic streams on ingress ports. Streams belonging to static MVR groups are always forwarded on MVR VLANs so that any host can join such channels immediately. However, dynamic groups are streamed from the server only if some host is interested in them. A command is provided to receive traffic on a port which is excluded by MVR. However, regular IGMP rules still apply to these ports, so the ports must have a router connected or an IGMP host to receive the stream.

These rules are to prevent multicast packets from leaking into an undesired virtual port, such as p2 on VLAN pc2 in Figure 150. These rules also allow that, in most topologies, MVR can be deployed with minimal configuration. However, unlike *EAPS* and *STP*

(Spanning Tree Protocol), MVR is not intended to be a Layer 2 protocol to solve packet looping problems. Since multicast packets leak across VLANs, one can misconfigure and end up with a multicast storm. MVR does not attempt to solve such problems.

Note
If a port is blocked by Layer 2 protocols, that port is removed from the egress list of the cache. This is done dynamically per the port state.

For most situations, you do not need to manually configure ports to receive the MVR multicast streams. But if one of the forwarding rules denies forwarding to a port that requires the streams, you can manually receive the MVR multicast streams by using the following command:

```
configure mvr vlan vlan_name add receiver port port-list
```

Inter-Multicast VLAN Forwarding

In Basic MVR Deployment, only simple topologies are considered, in which subscribers on different *VLAN*s access a multicast VLAN. There are topologies where streams need to be forwarded onto another multicast VLAN, as shown in Figure 151. In this figure, a Multicast Service Provider (MSP) multicast VLAN is attached to ports 1:1-2 on both switches, SW1 and SW2. On the customer side, another multicast VLAN, delivers multicast streams to other switches around the ring.

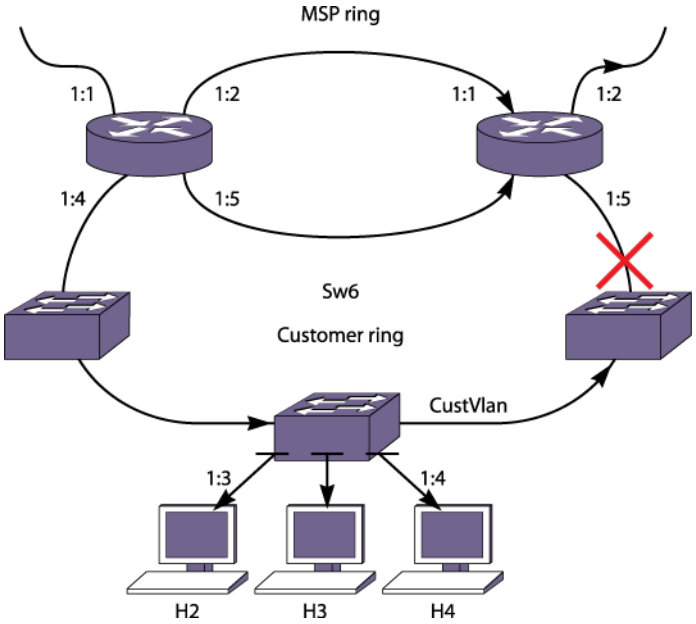


Figure 151: Inter-Multicast VLAN Forwarding

In this topology, a multicast stream can be leaked into the customer multicast network through either switch SW1 or SW2. However, as described in MVR Forwarding on page 1204, packets are not forwarded to router ports (ports 1:4 and 1:5 can be router ports if SW2 is an *IGMP* querier). To get around this, MVR needs to be configured on CustVlan either on SW1 or SW2. Since the forwarding rules apply only to non-MVR VLANs, traffic from one MVR VLAN is leaked into the router ports of another VLAN, if MVR is enabled on that.

In the topology above, the MSP multicast VLAN is carried on two switches that also carry the customer multicast VLAN. When multiple switches carry both multicast VLANs, it is imperative that MVR is configured on only one switch. Only that switch should be used as the transit point for multicast streams from one multicast ring into another. Otherwise, duplicate packets are forwarded. Also on the non-MVR switches, the ring ports should be configured as static router ports, so that ring ports are excluded from forwarding packets onto the customer ring. There is no mechanism to elect a designated MVR forwarder, so it must be configured correctly.

MVR Configurations

MVR enables Layer 2 network installations to deliver bandwidth intensive multicast streams. It is primarily aimed at delivering IPTV over Layer 2 networks, but it is valuable in many existing *EAPS* or *STP* installations. This section explores a few possible deployment scenarios and configuration details. Of course, real world networks can be lot different from these examples. This section is meant to present some ideas on how to deploy MVR over existing networks, as well as to design new networks that support MVR.

MVR with EAPS

Since MVR is designed with a Layer 2 ring topology in mind, it is strongly recommended that it should be deployed with *EAPS*. The MVR plus EAPS combination provides a superior solution for any triple play network, where the service provider intends to provide data, voice, and video services. EAPS is a proven solution for providing sub-second SONET-like protection to Layer 2 rings. For more detail on EAPS refer to EAPS.

Consider a typical EAPS topology in Figure 152, where 3 VLANs on the core ring serve various clients on each switch. To provide video service, one of the VLANs on the EAPS ring is designated as a multicast VLAN. MVR is enabled only on this VLAN (mcastvlan). V1 is the control VLAN, and V2 is another protected VLAN. A router serving the multicast feed would typically run PIM on mcastvlan, to support the static and dynamic *IGMP* membership on the VLAN.

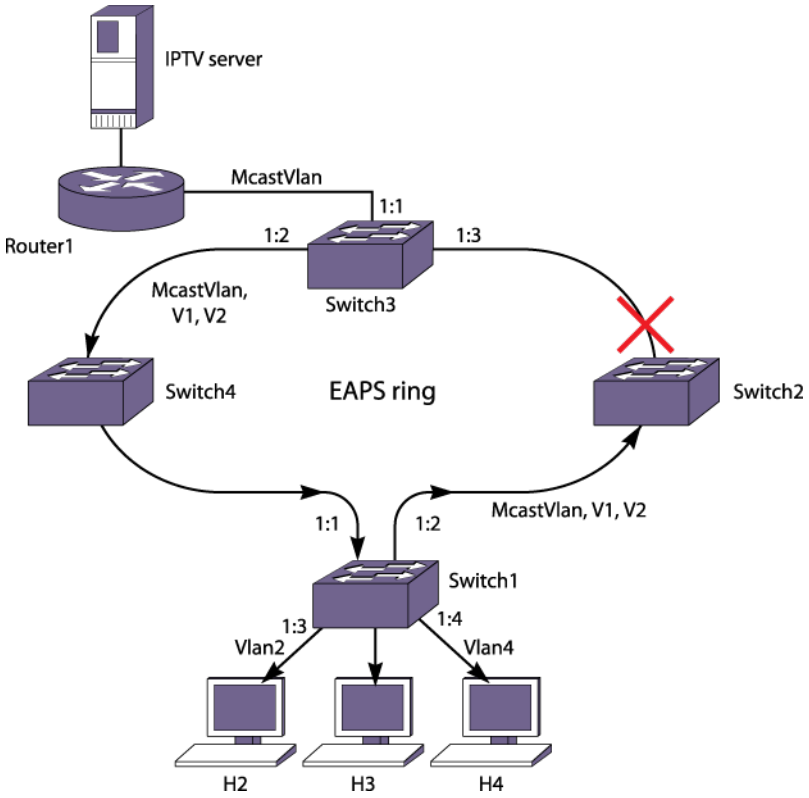


Figure 152: MVR on an EAPS Ring

The following is a typical configuration for the router and switches.

Router1:

```
create vlan mcastvlan
configure mcastvlan add port 1:1
create vlan server
configure server add port 1:2
configure mcastvlan ipaddress 10.1.1.1/24
configure server ipaddress 11.1.1.1/24
configure igmp snooping mcastvlan port 1:1 add static group 239.1.1.1
enable ipforwarding
enable ipmcforwarding
configure igmp snooping leave-timeout 2000
configure pim add vlan all
enable pim
```

Switch1:

```
create vlan mcastvlan
create vlan v1
create vlan v2
create vlan vlan2
configure vlan vlan2 add port 1:3
configure vlan vlan2 ipaddress 10.20.1.1/24
configure mcastvlan tag 20
configure mcastvlan add port 1:1,1:2 tag
configure mvr add vlan mcastvlan
configure vlan v1 tag 30
configure v1 add port 1:1,1:2 tag
configure vlan v2 tag 40
configure v2 add port 1:1,1:2 tag
```

```
create eaps e1
configure eaps e1 mode transit
configure eaps e1 add control vlan v1
configure eaps e1 add protect vlan mcastvlan
configure eaps e1 add protect vlan v2
configure eaps port primary port 1:1
configure eaps port secondary port 1:2
enable eaps
enable mvr
```

Switch3:

```
create vlan McastVlan
create vlan v1
create vlan v2
configure mcastvlan tag 20
configure mcastvlan add port 1:2,1:3 tag
configure mcastvlan add port 1:1
configure mvr add vlan mcastvlan
configure vlan v1 tag 30
configure v1 add port 1:2,1:3 tag
configure vlan v2 tag 40
configure v2 add port 1:2,1:3 tag

create eaps e1
configure eaps e1 mode master
configure eaps e1 add control vlan v1
configure eaps e1 add protect vlan mcastvlan
configure eaps e1 add protect vlan v2
configure eaps port primary port 1:3
configure eaps port secondary port 1:2
enable eaps
enable mvr
```

Note
In this example, Switch3 is the EAPS master, but any of the other switches in the ring could have been configured as the master.

MVR with STP

In a Layer 2 ring topology, MVR works with STP as it works with *EAPS*. However, in other Layer 2 topologies, additional configuration steps may be needed to make sure that multicast feeds reach all network segments. Extra configuration is required because all ports in the *VLAN* are part of an *STP* domain, so that solely by examining the configuration it is not clear whether a port is part of bigger ring or is just serving a few hosts. In EAPS this problem is solved by distinguishing between configured primary or secondary ports from other VLAN ports. Consider a simplified Layer 2 STP network as shown in Figure 153.

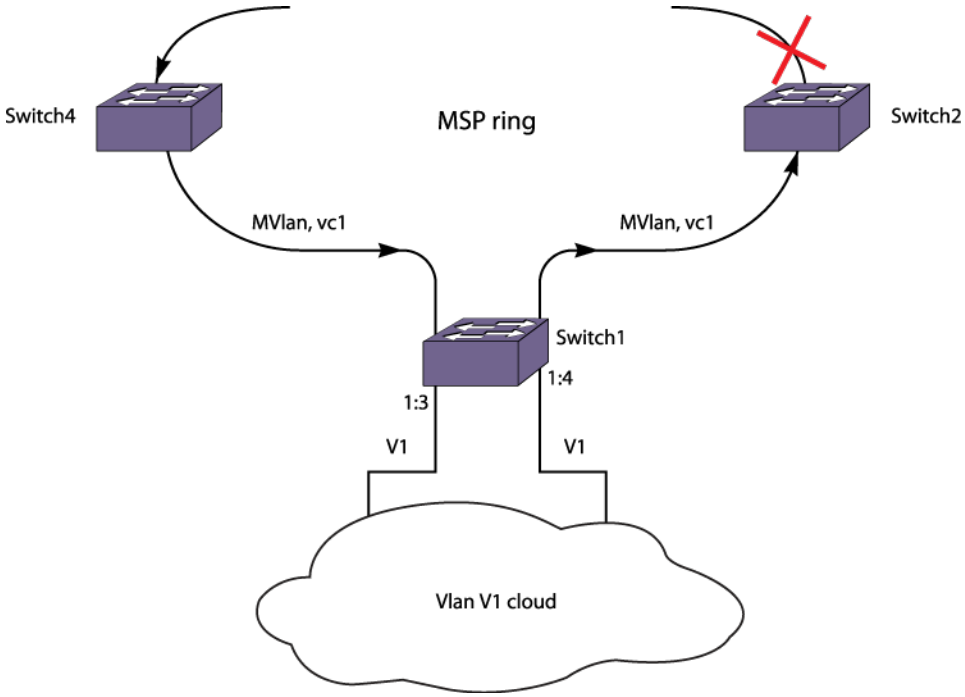


Figure 153: MVR with STP

In this topology, subscribers are in a Layer 2 cloud on VLAN V1.

STP is configured for all ports of V1. Since V1 spans on the ring as well, multicast cannot be forwarded on V1 blindly. Forwarding rules (described in MVR Forwarding on page 1204), dictate that multicast traffic is not forwarded on STP enabled ports. This is to make sure that multiple copies of multicast packets are not forwarded on the ring. However, since other STP enabled ports on V1 (1:3,1:4) are not part of the ring multicast stream, they need to be configured so that they get the packets. To configure the ports to receive packets, use the following command (mentioned in MVR Forwarding on page 1204):

```
configure mvr vlan vlan-name add receiver port port-list
```

Note
If the Layer 2 cloud is connected back to ring ports, traffic may end up leaking into VLAN V1 in the ring. There is no way to avoid that. So, such topologies must be avoided.

The following is a typical configuration for Switch 1:

```
create vlan v1
configure v1 tag 200
configure v1 add port 1:1, 1:2 tag
configure v1 add port 1:3, 1:4
create vlan mvlan
configure mvlan add port 1:1, 1:2
configure mvr add vlan mvlan
create stpd stp1
configure stp1 add vlan v1 port all
enable stpd stp1 port all
```

```
configure mvr vlan v1 add receiver port 1:3,1:4
enable mvr
```

MVR in a VMAN Environment

In the case of a VMAN, a packet is tagged with a VMAN tag in addition to a possible *VLAN* tag. This is to provide VLAN aggregation for all customer traffic in the VMAN ring. Each customer is given its own VLAN, and traffic from all customers can be tunneled on a single VMAN tag into the metro ring to an outside Broadband Remote Access Server (BRAS). In a VMAN network, multicast traffic can be distributed over a separate VLAN in the metro core. These packets are not subjected to VMAN tunneling. Thus, IPTV service can be provided on this multicast VLAN on a VMAN network.

MVR deployment in a VMAN environment is not any different from that in an *EAPS* environment, since a separate multicast VLAN on the metro ring is used for multicasting. However, it provides interesting capabilities to MSPs for their video offerings. Different service bundles can be offered on separate VLANs. Packets are not forwarded to any metro link segments where a stream is not required.

Figure 154 illustrates an example design for MVR in a VMAN environment. Any multicast packet entering on MVlan is forwarded on MVlan to the next switch. These multicast packets are not tunneled.

With MVR, switches on the VMAN do not have to run any routing protocol. If MVR is enabled on the multicast VLAN, MVlan, traffic is pulled from the IPTV server. Such multicast packets egressing from the CE port are always untagged. The downstream DSLAM distributes untagged multicast packets to the respective subscriber VLANs.

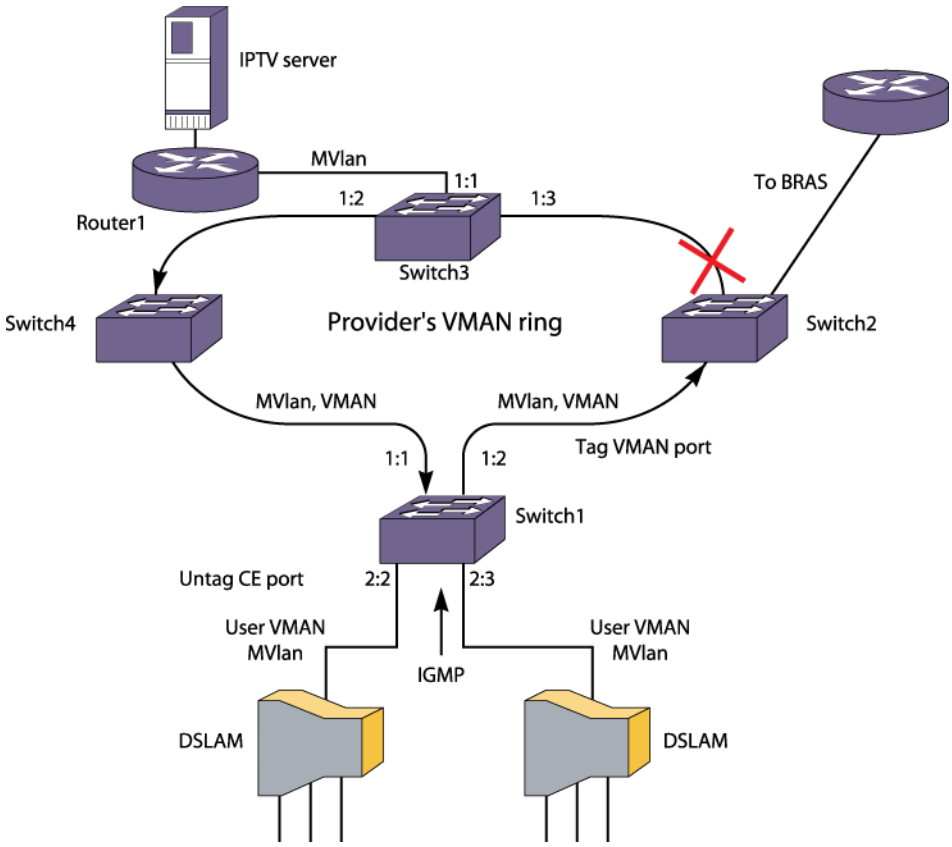


Figure 154: MVR in a VMAN Environment

The following is a typical configuration for Switch 1 in the above figure:

```
create vman vman2
configure vman vman2 tag 300
configure vman vman2 add port 2:2-2:3 untagged
configure vman vman2 add port 1:1,1:2 tagged
enable port 2:*
enable port 1:*
create vlan mvlan
configure vlan mvlan tag 200
configure vlan mvlan add port 1:1,1:2 tag
configure mvr add vlan mvlan
enable mvr
```

Displaying Multicast Information

Displaying the Multicast Routing Table

- To display part or all of the entries in the multicast routing table, use the following command:

  show iproute {**ipv4**} {{**vlan**} *name* | [*ipaddress netmask* | *ipNetmask*] | **origin** [**direct** | **static** | **mbgp** | **imbgp** | **embgp**]} **multicast** {**vr** *vr_name*}

Displaying the Multicast Cache

- The multicast cache stores information about multicast groups. To display part or all of the entries in the multicast cache, use the following command:

  ```
  show mcast cache {{vlan} vlan_name} {{[group grpaddressMask |
  grpaddressMask] {source sourceIP | sourceIP}} {type [snooping | pim |
  mvr]}| {summary}}
  ```

Looking Up a Multicast Route

- To look up the multicast route to a specific destination, use the following command with the **multicast** option:

  ```
  rtlookup [ipv4_address | ipv6_address] { unicast | multicast | rpf }
  {vr vr_name}
  ```

Looking Up the RPF for a Multicast Source

- To look up the RPF for a multicast source, use the following command with the **rpf** option:

  ```
  rtlookup [ipv4_address | ipv6_address] { unicast | multicast | rpf }
  {vr vr_name}
  ```

Displaying the PIM Snooping Configuration

- To display the PIM snooping configuration for a *VLAN*, use the following command:

  ```
  show pim snooping {vlan} vlan_name
  ```

Displaying Anycast Rendezvous Point (RP) Using PIM (RFC 4610)

To display rendezvous point (RP) for Anycast RP using PIM (RFC 4610) information, use the following command:

```
show pim {ipv4 | ipv6} anycast-rp {ip_address}
```

Troubleshooting PIM

Multicast Trace Tool

The multicast trace tool is the multicast equivalent of unicast trace route mechanism and is an effective tool for debugging multicast reachability problems. This tool is based on an IETF draft and uses *IGMP*. Because it is harder to trace a multicast path from the source to the destination, a multicast trace is run from the destination to the source. The multicast trace can be used to do the following:

- Locate where a multicast traffic flow stops
- Identify sub-optimal multicast paths

A multicast trace is used for tracing both potential and actual multicast forwarding tree paths. When the multicast tree is established and traffic is flowing, this tool traces the

actual traffic flow. If there is no traffic while executing the command, this tool displays the potential path for the group and source being traced.

You can direct a multicast trace to any network destination, which can be a multicast source or destination, or a node located between a source and destination. After you initiate the trace, a multicast trace query packet is sent to the last-hop multicast router for the specified destination. The query packet contains the source address, group address, destination/receiver address, response address, maximum number of hops, and TTL to be used for the multicast trace response.

The previous hop router selection is based on the multicast routing protocol and the state for the S,G entry in the processing router.

For example:

- If there is no S,G state in the router, the parent closest to the RP is chosen as the previous hop.
- If the S,G state exists in the router, the parent closest to the source is chosen as the previous hop.

The last hop router converts the multicast trace query into a unicast traceroute request by appending response data (for the last hop router) into the received query packet, and the last hop router forwards the request packet to the router that it believes is the proper previous hop for the given source and group.

Each multicast router adds its response data to the end of the request packet, and then forwards the modified unicast request to the previous hop.

The first hop router (the router that determines that packets from the source originate on one of its directly connected networks) changes the packet type to response packet and sends the completed response to the query generator. If a router along the multicast route is unable to determine the forwarding state for a multicast group, that router sends a response back to the originator with NO ROUTE forwarding code.

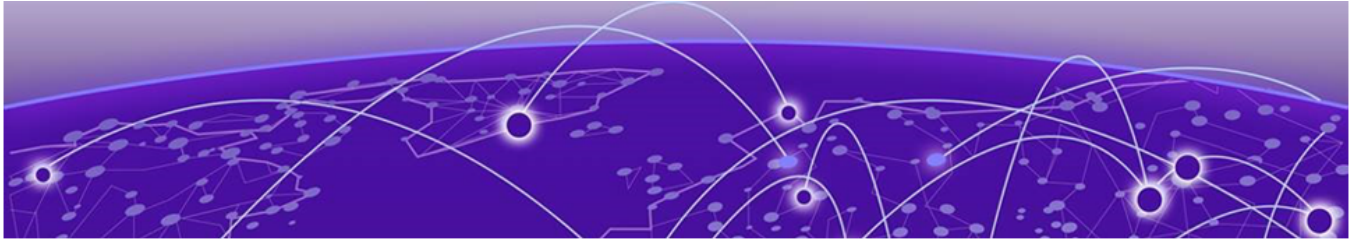
To initiate a multicast trace, use the following command:

```
mtrace source src_address {destination dest_address} {group grp_address}
{from from_address} {gateway gw_address} {timeout seconds} {maximum-hops
number} {router-alert [include | exclude]} {vr vrname}
```

Multicast Router Information Tool

The multicast router information tool is an ExtremeXOS command that allows you to request information from a specific multicast router. For more information, see the command description for the following command:

```
mrinfo {router_address} {from from_address} {timeout seconds} {multiple-
response-timeout multi_resp_timeout} {vr vrname}
```



Network Login

Network Login Overview on page 1214
Configuring Network Login on page 1229
Authenticating Users on page 1232
Local Database Authentication on page 1232
802.1X Authentication on page 1236
Web-Based Authentication on page 1246
MAC-Based Authentication on page 1254
Additional Network Login Configuration Details on page 1260

This chapter offers information about Network Login procedures. This section provides an overview, specific configuration information, and specific information about Local Database Authentication, 802.1X Authentication, Web-based Authentication, and MAC-based Authentication.

Network Login Overview

Network login controls the admission of user packets into a network by allowing MAC addresses from users that are properly authenticated. Network login is controlled on a per-port basis.

When network login is enabled on a port, that port does not forward any packets until authentication takes place.

Unknown broadcast/unicast/multicast (BUM) traffic is not allowed on the egress side of network login-enabled ports until authentication is successful. To allow BUM traffic to egress on network login-enabled, unauthenticated ports, use `configure netlogin` **`ports`** [*`port_list`* | **`all`**] **`allow egress-traffic`** [**`none`** | **`unicast`** | **`broadcast`** | **`all_cast`**] .

Network login is capable of three types of authentication: web-based, MAC-based, and 802.1X. In addition, network login has two different modes of operation: campus mode and ISP mode. The authentication types and modes of operation can be used in any combination.

When web-based network login is enabled on a switch port, that port is placed into a non-forwarding state until authentication takes place. To authenticate, a user must open a web browser and provide the appropriate credentials. These credentials are either approved, in which case the port is placed in forwarding mode, or not approved,

in which case the port remains blocked. You can initiate user logout by submitting a logout request or closing the logout window.

The following capabilities are included with network login:

- Web-based login using HTTP available on each port.
- Web-based login using HTTPS—if you install SSL—available on each port.
- Multiple supplicants for web-based, MAC-based, and 802.1X authentication on each port.

ExtremeXOS *NetLogin* provides the *AAA (Authentication, Authorization, and Accounting)* functionality, which is an important block of the network infrastructure and security and provides a model or framework to determine who is requesting network access, network resources that can be accessed by the requesting party, and when the resources are used. NetLogin supports all popular methods of authentication: MAC-based, Web-based, and IEEE 802.1X. NetLogin can help network administrators to control access into the network; it also provides flexibility to configure specific backend resources to which user access is allowed.

Together with IP Security, administrators can enhance security in the network by controlling access to upstream network and resources by the hosts or clients. IP Security is a collection of powerful features that allow network administrators to design security in combination with standard authentication and authorization techniques. IP Security features as ExtremeXOS 15.2 include:

- *DHCP (Dynamic Host Configuration Protocol)* Snooping and concept of trusted DHCP Servers
- Source IP Lockdown
- ARP Learning and Validation
- Gratuitous ARP Protection

When NetLogin and IP Security features are enabled on a port, NetLogin performs the first or the basic function of authenticating and authorizing the user. Further course of

action is determined by IP Security in case a violation is detected. The violation action will then determine further access to the network.

Scenario	Notes	Expected Behavior
NetLogin + DHCP Snooping and trusted DHCP Servers/Ports. Violation: DHCP Server Packets seen on netlogin enabled ports (i.e. a host is masquerading post authentication).	We recommend that you enable NetLogin on the client-facing ports. Enabling DHCP Snooping on all ports, including client/host facing ports and ports connected to the upstream network automation and provisioning infrastructure, ensures that all DHCP messages are processed by the switch and a DHCP binding database is maintained. For the combination of NetLogin and DHCP Security to work correctly, we recommend that you configure at least one uplink (or server facing port) as a trusted port. This ensures that all other ports (normally client-facing ports) automatically become untrusted, and can be monitored for any violations that might occur. You can configure more than one uplink port as "trusted" as this allows flexibility in network design. In addition to controlling which DHCP Servers can communicate with the downstream clients, the trusted DHCP Server configuration can be used.	Action: None NetLogin authenticates the client, and IP Security flags a violation. No action is taken because of configuration. Action: Drop-packet The packet is dropped, and an EMS event is logged. Action: block-mac NetLogin initially authenticates the client, and subsequently when the violation occurs, IP Security reports violation, which causes the MAC address to be blocked either permanently or for a configured duration on the switch. The *FDB (forwarding database)* will be flushed after FDB entry ages out and the netlogin entry is unauthenticated and removed from the switch. Action: block-port NetLogin initially authenticates the client, and subsequently when the violation occurs, IP Security reports the violation, which causes the port to be administratively disabled. As a result, all authenticated clients on the ports are immediately unauthenticated and removed from the switch. This can occur either for a certain configured duration or permanently. We do not recommend that you use this configuration if there are multiple supplicants on the port (for example, conference rooms, groups of users, etc. accessing the network through an intermediate port extender, or hub). In addition, for network troubleshooting and debugging purposes, an *SNMP (Simple Network Management*

Scenario	Notes	Expected Behavior
		Protocol) trap can be sent to a central network manager.
NetLogin + Source IP Lockdown Violation: After a client/host successfully authenticates to the network, it performs a source IP address violation.	The Source IP lockdown feature can determine if a client/host should be allowed access to the network based on inspection of the source IP address used in the packets. If a client is not using an IP address present in DHCP binding database, a violation is flagged for the client and further action is determined by configuration. This helps prevent clients from using source addresses not assigned by a centralized network automation and provisioning infrastructure. By default the switch denies all IP traffic from clients when source IP lockdown is enabled. In order for the clients to get a valid IP address, DHCP packets are allowed to be forwarded through the switch. NetLogin authentication (all three forms) will still proceed, and clients presenting the valid credentials (per authentication scheme) are authenticated. Post authentication (and authorization - for membership to a *VLAN (Virtual LAN)*), once a valid DHCP binding is found for the client, access control lists are automatically (dynamically) applied to permit traffic from the client.	It is not mandatory to configure specific violation actions in DHCP Snooping (which is a prerequisite to this feature). If configured, DHCP Snooping filters and violation actions take precedence over source IP lockdown. This is to ensure that successfully authenticated clients (with NetLogin) do not masquerade as rogue DHCP servers post authentication. In this case, DHCP violation is detected and actions are determined per configuration for vilation-action.

Scenario	Notes	Expected Behavior
NetLogin + DHCP Secured ARP	DHCP Secured ARP allows administrators to control how the ARP table is populated. By default, the switch learns IP ARP bindings and builds the ARP table by tracking the ARP requests and replies. When DHCP Secured ARP is used for design, IP ARP learning method can be disabled. This is recommended for security purposes. When combined with NetLogin, this feature ensures that a client (even after success authentication) cannot override the ARP entry on the switch, thereby preventing duplicate addresses and ensuring proper network operation. ARP entries populated from DHCP are known as Secure ARP entries and are flushed/removed only when the address is released.	Same behavior as expected in the case of "NetLogin + DHCP Snooping and Trusted Ports/Server". Apart from securing the IP ARP table, in this case, violations detected as part of DHCP Snooping and validation is flagged and actions are determined by configuration.

Scenario	Notes	Expected Behavior
NetLogin + ARP Validation	ARP Validation helps check different fields in the ARP packet. Source MAC, Destination MAC, and Source IP addresses can be checked for validity. For a complete reference to the different checks performed, please refer to ARP Validation Options. All checks are performed post authentication.	Violation actions are same as the options presented in the DHCP Snooping and Validation Cases. Same expected behavior and functionality as in the case of "NetLogin + DHCP Snooping and Trusted Ports/Server".
NetLogin + Gratuitous ARP Protection	Gratuitous ARP is a method by which a client/host can resolve its own IP address, and is useful in scenarios where duplicate addresses need to be detected, or a host can announce that it has either used an IP address on a different NIC card, or even if a client has moved from one location to another. While ExtremeXOS supports Gratuitous ARP, protection can also be enabled to mitigate any risk or threats that can arise because of any m-i-m attacks. The switch will automatically send ARP packets to not only protect its own IP address but to also safeguard addresses of any NetLogin authenticated clients on the switch. For this, it is recommended that both DHCP Secured ARP and Gratuitous ARP protection be enabled on the switch. It is not mandatory that DHCP Snooping be enabled for this feature, but becomes a prerequisite if DHCP Secured ARP is also configured.	ARP Packets are sent out when a violation is detected. For all other violations detected by DHCP Snooping and Trust, the corresponding violation actions are determined by configuration, and expected behavior is the same as the case of "NetLogin + DHCP Snooping and Trusted Servers/Ports".

Web-Based, MAC-Based, and 802.1X Authentication

Authentication is handled as a web-based process, MAC-based process, or as described in the IEEE 802.1X specification.

Web-based network login does not require any specific client software and can work with any HTTP-compliant web browser. By contrast, 802.1X authentication may require additional software installed on the client workstation, making it less suitable for a user walk-up situation, such as a cybercafé or coffee shop. A workstation running Windows 7

or Windows 8 supports 802.1X natively, and does not require additional authentication software. Extreme Networks supports a smooth transition from web-based to 802.1X authentication.

Note
When both HTTP and HTTPS are enabled on the switch and sending HTTP requests from the Netlogin client, HTTPS takes preference and the switch responds with a HTTPS response.

MAC-based authentication is used for supplicants that do not support a network login mode, or supplicants that are not aware of the existence of such security measures, for example an IP phone.

If a MAC address is detected on a MAC-based enabled network login port, an authentication request is sent once to the AAA application. AAA tries to authenticate the MAC address against the configured Remote Authentication Dial In User Server (*RADIUS (Remote Authentication Dial In User Service)*) server and its configured parameters (timeout, retries, and so on) or the configured local database.

The credentials used for this are the supplicant's MAC address in ASCII representation and a locally configured password on the switch. If no password is configured, the MAC address is also used as the password. You can also group MAC addresses together using a mask (`configure netlogin add mac-list [mac {mask} | default] {encrypted {encrypted_password | password} {ports port_list}`).

DHCP is required for web-based network login because the underlying protocol used to carry authentication request-response is HTTP. The client requires an IP address to send and receive HTTP packets before the client is authenticated; however, the only connection that exists is to the authenticator. As a result, the authenticator must be furnished with a temporary DHCP server to distribute the IP address.

The switch responds to DHCP requests for unauthenticated clients when DHCP parameters such as *dhcp-address-range* and *dhcp-options* are configured on the network login *VLAN*. The switch can also answer DHCP requests following authentication if DHCP is enabled on the specified VLAN. If network login clients are required to obtain DHCP leases from an external DHCP server elsewhere on the network, DHCP should not be enabled on the VLAN.

Also, enabling DHCP on post authentication VLANs is not be saved in the switch configuration, since the port movement is dynamic. The following warning message appears when enabling DHCP on post authentication VLAN and network login VLAN:

```
Warning: DHCP server configuration will not be saved for netlogin-enabled ports: 1

After reboot/port removal the dhcp config should be reconfigured again
```

The DHCP allocation for network login has a short time duration of 10 seconds and is intended to perform web-based network login only. The Netlogin lease timer can be extended using the command: `configure vlan vlan_name` **`netlogin-lease-timer`** *`seconds`* . As soon as the client is authenticated, it is deprived of this address. The client must obtain an operational address from another DHCP server in the network. DHCP is

not required for 802.1X, because 802.1X uses only Layer 2 frames (EAPOL) or MAC-based network login.

URL redirection (applicable to web-based mode only) is a mechanism to redirect any HTTP request to the base URL of the authenticator when the port is in unauthenticated mode. In other words, when the user tries to log in to the network using the browser, the user is first redirected to the network login page. Only after a successful login is the user connected to the network. URL redirection requires that the switch is configured with a DNS client.

Web-based, MAC-based, and 802.1X authentication each have advantages and disadvantages, as summarized in Advantages of Web-Based Authentication.

Advantages of Web-Based Authentication:

- Works with any operating system that is capable of obtaining an IP address using DHCP. There is no need for special client side software; only a web browser is needed.

Disadvantages of Web-Based Authentication:

- The login process involves manipulation of IP addresses and must be done outside the scope of a normal computer login process. It is not tied to a Windows login. The client must bring up a login page and initiate a login.
- Supplicants cannot be re-authenticated transparently. They cannot be re-authenticated from the authenticator side.
- This method is not as effective in maintaining privacy protection.

Advantages of MAC-Based Authentication:

- Works with any operating system or network enabled device.
- Works silently; the user, client, or device does not know that it gets authenticated.
- Ease of management - set of devices can easily be grouped by the vendor part of the MAC address.

Disadvantages of MAC-Based Authentication:

- Security is based on the MAC address of the client, so the network is more vulnerable to spoofing attacks.

Advantages of 802.1X Authentication:

- In cases where the 802.1X is natively supported, login and authentication happens transparently.
- Authentication happens at Layer 2. It does not involve getting a temporary IP address and subsequent release of the address to obtain a permanent IP address.
- Allows for periodic, transparent re-authentication of supplicants.

Disadvantages of 802.1X Authentication:

- 802.1X native support is available only on newer operating systems, such as Windows 7 or Windows 8.

- 802.1X requires an EAP-capable RADIUS Server. Most current RADIUS servers support EAP, so this is not a major disadvantage.
- Transport Layer Security (TLS) and Tunneled TLS (TTLS) authentication methods involve Public Key Infrastructure (PKI), which adds to the administrative requirements.

Multiple Supplicant Support

An important enhancement over the IEEE 802.1X standard is that ExtremeXOS supports multiple supplicants (clients) to be individually authenticated on the same port.

This feature makes it possible for two or more client stations to be connected to the same port, with some being authenticated while others are not. A port's authentication state is the logical “OR” of the individual MAC's authentication states. In other words, a port is authenticated if any of its connected clients is authenticated. Multiple clients can be connected to a single port of authentication server through a hub or Layer 2 switch.

Multiple supplicants are supported in ISP mode for web-based, 802.1X, and MAC-based authentication. In addition, multiple supplicants are supported in Campus mode if you configure and enable network login MAC-based VLANs. For more information, see Configuring Network Login MAC-Based VLANs on page 1260.

The choice of web-based versus 802.1X authentication is again on a per-MAC basis. Among multiple clients on the same port, it is possible that some clients use web-based mode to authenticate, and some others use 802.1X, but the restriction is that they must be in the same untagged *VLAN*. This restriction is not applicable if you configure network login MAC-based VLANs. For more information, see Configuring Network Login MAC-Based VLANs.

Note

With multiple supplicant support, after the first MAC is authenticated, the port is transitioned to the authenticated state and other unauthenticated MACs can listen to all data destined for the first MAC. Be aware of this as unauthenticated MACs can listen to all broadcast and multicast traffic directed to a network login-authenticated port.

Network Login Multiple Authentication Support

The client or supplicant connected to the NetLogin-enabled port(s) are authenticated by only one authentication protocol. If enabled globally and at the port, MAC-based authentication takes precedence if enabled globally and at the port. Dot1x takes precedence over MAC-based authentication if Dot1x is supported by the supplicant. In this case the MAC-based authentication information is cleared as the client gets authenticated via Dot1x. Web-based authentication happens only when the port belongs to the NetLogin *VLAN*. The final authentication method used with its associated actions is applied while any previous authenticated protocol information is cleared.

This feature supports multiple authentication protocols on a NetLogin-enabled port. The user must specify the authentication protocol priority or order per port which dictates the action for the client or supplicant that is getting authenticated on this port. Use the CLI to configure the authentication protocol order (`configure netlogin **authentication protocol-order** [[**dot1x** [**web-based** | **mac** | **cep**]] | [**mac** [**dot1x** | **web-based** | **cep**]] | [**web-based** [**dot1x** | **mac** | **cep**]] | [**cep** [**dot1x** | **web-based** | **mac**]]]`). By default the protocol precedence order for a NetLogin-enabled port is:

1. Dot1x
2. Web-based
3. MAC
4. CEP

For example, if the following is the authentication protocol order configured on a NetLogin-enabled port in which three authentication protocols are enabled:

1. Dot1x
2. MAC
3. Web-based

Note
Precedence order does not work when MAC and web-based are enabled on the same port. If you want to authenticate by web-based, do not use MAC and other protocols.

When user “john” tries to authenticate with his login credentials through Dot1x-enabled supplicant or client, it sends the EAPOL packet to the ExtremeXOS switch or authenticator. Upon receipt of the EAPOL packet, the ExtremeXOS kernel *FDB* module informs the user interface FDBMgr about the new MAC detection. The FDBMgr in turn informs the NetLogin process about the new MAC or client. The NetLogin process tries to authenticate the client/MAC through *RADIUS*. On receiving the authentication result from AAA process, the NetLogin process checks for the protocol precedence configured by the user for that port and also finds if this client is being authenticated by any other authentication protocol. In this case, no other authentication protocol has authenticated this MAC address yet and the NetLogin process applies the action (VLAN movement, UPM security profile, etc.;) corresponding to MAC-based authentication.

The ExtremeXOS switch or authenticator then sends the credentials of user “john” to the authentication server (RADIUS) a second time for Dot1x protocol authentication, After the authentication result is received, the NetLogin process again checks the protocol precedence to find that the user “john's" host/MAC is already authenticated using MAC-based authentication. Since Dot1x is configured as the highest precedence protocol for this port the NetLogin process remove MAC-based authentication actions for this client and apply the Dot1x protocol action for “john” on this port. The MAC-based authenticated client continues to exist and performs the periodic re-authentication for the configured time. The `show netlogin` output shows the client’s highest precedence protocol or action applied authentication protocol details only.

When another user “sam” tries to authenticate from the same host or MAC through web-based authentication method (provided the NetLogin-enabled port is still

present in NetLogin VLAN) the user “sam” gets authenticated, but the web-based authentication protocol action is not applied, since user “john” is already authenticated from this MAC address with user-configured highest precedence Dot1x protocol on this port.

Note
After changing the protocol precedence, the action for the current highest precedence protocol (if client is authenticated by this protocol) takes effect immediately.

Note
After disabling the highest precedence protocol on this port, the next precedence protocol (if client is authenticated by this protocol) action takes effect immediately.

Support for Attaching and Detaching the UPM profile

When the user or device gets authenticated the netlogin process will check for the protocol precedence configured by the user in this port and apply or remove the action accordingly. From the previous example when the user “john” tries to authenticate using Dot1x the ExtremeXOS switch or authenticator will authenticate the MAC using MAC based authentication and apply the action corresponding to the MAC based authentication protocol which includes applying UPM profile, creating *VLAN* (if netlogin dynamic VLAN is enabled on this port), VLAN movement etc.; when the user “john” then gets authenticated through Dot1x the ExtremeXOS switch or authenticator determines that Dot1x is the highest precedence protocol configured by the user in this port and removes the actions of MAC based authentication protocol and applies the Dot1x authentication protocol action that includes applying UPM profile, creating VLAN (if netlogin dynamic VLAN is enabled on this port), VLAN movement etc.; the MAC based authenticated client details still remains and continues to get reauthenticated for the configured time.

The netlogin process does the following when the user or MAC is being unauthenticated:

1. Sends accounting stop message to *RADIUS* through AAA.
2. Logs unauthentication EMS message for this client for this authentication protocol.
3. Sends “extremeNetloginUserLogout” SNMP trap message for this authentication protocol.
4. Informs IDM about unauthentication of this client for this authentication protocol.
5. Informs UPM about the client’s unauthentication for this protocol.

After performing the above actions, the netlogin process applies the highest precedence authentication protocol action configured for this port.

Campus and ISP Modes

Network login supports two modes of operation, Campus and ISP.

Campus mode is intended for mobile users who tend to move from one port to another and connect at various locations in the network. ISP mode is meant for users who connect through the same port and *VLAN* each time (the switch functions as an ISP).

In Campus mode, the clients are placed into a permanent VLAN following authentication with access to network resources. For wired ports, the port is moved from the temporary to the permanent VLAN.

In ISP mode, the port and VLAN remain constant. Before the supplicant is authenticated, the port is in an unauthenticated state. After authentication, the port forwards packets.

You do not explicitly configure the mode of operation; rather, the presence of any Extreme Networks Vendor Specific Attribute (VSA) that has a VLAN name or VLAN ID (any VLAN attribute) in the *RADIUS* server determines the mode of operation. If a VLAN attribute is present, it is assumed to be Campus mode. If a VLAN attribute is not present, it is assumed to be ISP mode.

Note

When a client is authenticated in multiple VLANs using campus mode: 1) If any of the authenticated VLANs are deleted manually from a port or globally, the client is unauthenticated from all VLANs; and 2) If traffic is not seen on a particular VLAN then the *FDB* entry ages out and is deleted; the client itself remains authenticated and the FDB entry is reinstalled either when traffic is detected on that VLAN or when the client reauthenticates. For additional information on Campus and ISP mode operation on ports that support network login and *STP (Spanning Tree Protocol)*, see Exclusions and Limitations.

Spanning Tree Protocol (STP) Support with NetLogin Campus Mode

ExtremeXOS provides support for *STP* with NetLogin campus mode.

If NetLogin is authenticating a client in campus mode on a port, then that port is moved/added to list of authentication *VLAN*s (provided in VLAN VSA), and as a consequence the port could be removed from the VLANs where it is currently present. Due to this constant port/VLAN movement in campus mode, any STP configuration on the port could be lost; so to protect the port by STP, autobind needs to be configured on the list of authentication VLANs provided in VLAN VSA.

For example, configure RSTP with autobind on VLANs V1, V2, and V3 assuming that there are no ports associated to any of those VLANs. NetLogin moves the ports to the VLANs as when clients get authenticated. As the ports are added to the VLAN, RSTP inherits only tagged ports or untagged ports, depending on the default encapsulation configured for the S1 domain. If S1 is configured with default-encapsulation PVST+ or EMISTP, and if NetLogin adds port P1 to V1 (tagged), then RSTP inherits port P1. This occurs dynamically, and if port P1 is removed from VLAN V1, then port P1 is automatically removed from S1.

This feature is generic and applicable to all STP operational modes.

Supported Platforms

All ExtremeSwitching switches.

Network Login and Hitless Failover

When redundancy is available in a SummitStack, one node assumes the role of primary and the another node assumes the role of backup node.

The primary node executes the switch's management functions, and the backup node acts in a standby role. Hitless failover transfers switch management control from the primary node to the backup node.

Note
Not all platforms support hitless failover in the same software release. To verify if the software version you are running supports hitless failover, see Protocol Support for Hitless Failover on page 90. For more information about protocol and platform support for hitless failover, see Understanding Hitless Failover Support on page 89.

Network login supports hitless failover by relaying current client authentication information from the master node to the backup node. For example, if a client moves to the authenticated state, or moves from an authenticated state to an unauthenticated state, the primary node conveys this information to the backup node. If failover occurs, your authenticated client continues to operate as before the failover.

Backup nodes do not show reauthentication timer in the output of the show netlogin command as in master. Only the authenticated clients details appear in show netlogin in backup.

Note
If you use 802.1X network login, authenticated clients remain authenticated during failover; however, shortly after failover, all authenticated clients automatically re-authenticate themselves. Re-authentication occurs without user intervention.

If failover occurs during the authentication or re-authentication of a client, the client must repeat the authentication process.

Note
Before initiating failover, review the section Synchronizing Nodes on page 1721 to confirm that your switch (or SummitStack) and both (or all) nodes are running software that supports the synchronize command.

Initial Hitless Failover

To initiate hitless failover on a network that uses network login:

1. Confirm that the nodes are synchronized and have identical software and switch configurations using the show switch {detail} command.

   If the primary and backup nodes, are not synchronized and both nodes are running a version of ExtremeXOS that supports synchronization, proceed to the next step.

   If the primary and backup nodes, are synchronized, proceed to Step 3 on page 1227.

   The output displays the status of the primary and backup nodes, with the primary node showing MASTER and the backup node showing BACKUP (InSync).
2. If the primary and backup nodes, are not synchronized, use the synchronize command to replicate all saved images and configurations from the primary to the backup.

   After you confirm that the nodes are synchronized, proceed to Step 3.
3. If the nodes are synchronized, use the `show ports tdm information` command to initiate failover.

For more detailed information about verifying the status of the nodes and system redundancy, see Understanding System Redundancy on page 84. For more information about hitless failover, see Understanding Hitless Failover Support on page 89.

Network Login over LAG

When *NetLogin* is enabled on a sharing group, user credentials for the *LAG (Link Aggregation Group)* are forwarded to AAA server for authentication. Once authenticated, the LAG is moved to the *VLAN* configured as the destination VLAN for NetLogin, and the learned MAC address is installed over the LAG in the *FDB* on the NetLogin's destination VLAN.

All NetLogin configurations should be done on the LAG master port. For example:

```
enable sharing 24 grouping 24, 25, 26

enable netlogin ports 24 mac
```

For MAC-based authentication (see MAC-Based Authentication on page 1254), when NetLogin is enabled on a sharing group, software-based learning is enabled for each

member port of the sharing group. Similarly, when member ports are removed from the sharing group, software-based learning is disabled on that member port.

Note

- When a LAG is removed, all the NetLogin configurations related to that LAG are removed. Before deleting a sharing group, disable NetLogin on the LAG port.
- The master port cannot be removed from the LAG.
- The maximum number of authenticated users per LAG group is 1,024.
- If OnePolicy is enabled, NetLogin global protocol configurations and NetLogin VLAN configurations are lost, and then the LAG port is authenticated using OnePolicy by enabling NetLogin protocols globally.

Network Login over MLAG

Starting with ExtremeXOS 22.4, NetLogin MAC-based authentication and 802.1X authentication are supported in policy mode (NetLogin web-based authentication is not supported) over ports. If an MLAG peer goes down, supplicants remain authenticated on the other MLAG peers and continue to send data traffic thereby providing redundancy.

After successful authentication, NetLogin passes the necessary information to the policy module (such as the port, MAC address, policy profile, and authentication result) for further processing, such as movement of the port to VLAN and insertion of MAC address in the FDB.

Note
ISC port must be added to the destination VLAN for all dynamic VLANs on the switch.

Limitations

- Web-based authentication is not supported.
- Supported in policy mode only.

MLAG Support for Change of Authentication (CoA)

Starting with ExtremeXOS 30.4, MLAG support for Netlogin is extended for dynamic authorization and disconnect (CoA). CoA support (introduced in ExtremeXOS 22.1), which was implemented directly in the policy module, now moves to NetLogin.

The CoA messages that were previously handled directly by the policy module are now handled by the NetLogin module. With ExtremeXOS 30.4, the AAA module, on receiving the CoA messages from the RADIUS server, passes the data to NetLogin. Netlogin checks for the availability of the client, and the mode of authentication. If it finds a valid client, the CoA message is passed to policy for further processing. Additionally, NetLogin takes care of updating the MLAG peer about the dynamic authorization changes or disconnect. The NetLogin module on the MLAG peer passes the message to the policy module of that switch.

Limitations

- CoA support is not provided for web-based clients.
- The CoA server is aware of only the MLAG authenticator, so if the ISC link is down, one MLAG peer might have a client in a different profile with respect to the other MLAG node. This is resolved after the ISC link becomes active. This is a limitation of the CoA server.

Configuring Network Login

This section provides a general overview of the commands used for:

- Enabling or Disabling Network Login on the Switch on page 1229
- Enabling or Disabling Network Login on a Specific Port on page 1230
- Configuring the Move Fail Action on page 1230
- Displaying Network Login Settings on page 1230
- Exclusions and Limitations on page 1231

This section also describes information about Network Login's Exclusions and Limitations on page 1231.

For more detailed information about a specific mode of network login, including configuration examples, see the following sections:

- 802.1X Authentication on page 1236
- Web-Based Authentication on page 1246
- MAC-Based Authentication on page 1254

Note
When *STP* with Edge-safeguard and the Network login feature are enabled on same port, the port moves to a disabled state when it detects a loop in the network.

Enabling or Disabling Network Login on the Switch

By default, network login is disabled. To enable or disable network login and specify the authentication method, use the following commands:

- `configure netlogin` **`vlan`** *`vlan_name`*
- `enable netlogin [{`**`dot1x`**`} {`**`mac`**`} {`**`web-based`**`}]`
- `disable netlogin [{`**`dot1x`**`} {`**`mac`**`} {`**`web-based`**`}]`

Enabling or Disabling Network Login on a Specific Port

Network login must be disabled on a port before you can delete a *VLAN* that contains that port.By default, all methods of network login are disabled on all ports.

- To enable network login on a port to specify the ports and the authentication method, use the following command:
  `enable netlogin ports ports [{dot1x} {mac} {web-based}]`

  Note
  When network login and *STP* are enabled on the same port, network login operates only when the STP port is in the forwarding state.

- To disable network login, use the following command:
  `disable netlogin ports ports [{dot1x} {mac} {web-based}]`

Configuring the Move Fail Action

If network login fails to perform Campus mode login, you can configure the switch to authenticate the client in the original *VLAN* or deny authentication even if the user name and password are correct. For example, this may occur if a destination VLAN does not exist.

To configure the behavior of network login if a VLAN move fails, use the following command:

`configure netlogin move-fail-action [authenticate | deny]`

By default, the setting is deny.

The following describes the parameters of this command if two clients want to move to a different untagged VLAN on the same port:

- **authenticate**—Network login authenticates the first client that requests a move and moves that client to the requested VLAN. Network login authenticates the second client but does not move that client to the requested VLAN. The second client moves to the first client's authenticated VLAN.
- **deny**—Network login authenticates the first client that requests a move and moves that client. Network login does not authenticate the second client.

The dot1x client is not informed of the VLAN move-fail because it always receives EAP-Success or EAP-Fail directly based on the authentication result, not based on both authentication and the VLAN move result.

Displaying Network Login Settings

To display the network login settings and parameters, use the following command:

`show netlogin {port portlist vlan vlan_name} {dot1x {detail}}`

Setting Netlogin Traps

To set NetLogin trap setting for ports, use the following command:

```
configure netlogin ports [all | port_list] [allowed-users allowed_users
| authentication mode [optional | required] | trap [all-traps | no-traps
| [{success} {failed} {terminated} {max-reached}]]]
```

To set NetLogin system traps, use the following command:

```
configure netlogin trap max-users [enable | disable]
```

Exclusions and Limitations

The following are limitations and exclusions for network login:

- When using *NetLogin* MAC-based *VLAN* mode, moving a port as untagged from a pre-authentication VLAN to a post-authentication VLAN is not supported when both VLANs are configured with Protocol Filter IP.
- All unauthenticated MACs will be seeing broadcasts and multicasts sent to the port if even a single MAC is authenticated on that port.
- Network login must be disabled on a port before that port can be deleted from a VLAN.
- In Campus mode on all switches with untagged VLANs and the network login ports' mode configured as port-based-VLAN, after the port moves to the destination VLAN, the original VLAN for that port is not displayed.
- A network login VLAN port should not be a part of following protocols:
  - *EAPS (Extreme Automatic Protection Switching)*
  - *ESRP (Extreme Standby Router Protocol)*
- Network login and *STP* operate on the same port as follows:
  - At least one VLAN on the intended port should be configured both for network login and STP.
  - When STP blocks a port, network login does not process authentication requests and BPDUs are the only traffic in and out of the port. All user data forwarding stops.
  - When STP places a port in forwarding state, network login operates and BPDUs and user data flow in and out of the port. The forwarding state is the only STP state that allows network login and user data forwarding.
  - If a network login client is authenticated in ISP mode and STP blocks one of the authenticated VLANS on a given port, the client is unauthenticated only from the port or VLAN which is blocked.
  - All clients that are going through authentication and are learned on a blocked port or VLAN are cleared.

  Note
  When STP with edge-safeguard and network login feature is enabled on the same port, the port goes into the disabled state after detecting a loop in the network. NetLogin campus mode and STP can be configured together on a port with the autobind feature.
- The maximum re-authentication period configured using CLI commands is 86,400 seconds for MAC and Dot1x authentication methods. There is no limitation for the

re-authentication period (session timeout) sent using RADIUS server for Netlogin authentication.

- When in non-policy mode, NetLogin is not supported on user VRs. In non-policy mode, it is supported only on VR-Default.

  Enabling NetLogin on ports that are not part of VR-default produces the following error
  ```
  WARNING: Ports that are not part of the current Virtual Router were ignored.
  ```
- When in policy mode, system-wide, only one NetLogin base VLAN can be configured and this VLAN can be part of any VR.
- If you want to scale to 65,000 authenticated users, use a session timeout value of at least 300 minutes.
- NetLogin is not supported on user VRs in non-policy mode. NetLogin is supported only on VR-Default in non-policy mode.

Authenticating Users

There are three methods to authenticate users who log in to the switch: *RADIUS* client, TACACS+, and a local database of accounts and passwords.

Note

You cannot configure RADIUS and TACACS+ at the same time.

Local Database Authentication

You can configure the switch to use its local database for web-based and MAC-based network login authentication.

802.1X network login does not support local database authentication. Local authentication essentially mimics the functionality of the remote *RADIUS* server locally. This method of authentication is useful in the following situations:

- If both the primary and secondary (if configured) RADIUS servers timeout or are unable to respond to authentication requests.
- If no RADIUS servers are configured.
- If the RADIUS server used for network login authentication is disabled.

If any of the above conditions are met, the switch checks for a local user account and attempts to authenticate against that local account.

For local authentication to occur, you must configure the switch's local database with a user name and password for network login. We recommend a maximum of 64 local accounts. If you need more than 64 local accounts, we recommend using RADIUS for authentication. You can also specify the destination *VLAN* to enter upon a successful authentication. You can also specify a UPM profile in the local database when creating a local user.

You can also use local database authentication in conjunction with network login MAC-based VLANs. For more detailed information about network login MAC-based VLANs, see Configuring Network Login MAC-Based VLANs on page 1260.

Creating a Local Network Login Account--User Name and Password Only

We recommend creating a maximum of 64 local accounts. If you need more than 64 local accounts, we recommend using *RADIUS* for authentication. For information about RADIUS authentication, see Configuring the RADIUS Client on page 1611.

User names are not case-sensitive. Passwords are case-sensitive. User names must have a minimum of one character and a maximum of 32 characters. Passwords must have a minimum of zero characters and a maximum of 32 characters. If you use RADIUS for authentication, we recommend that you use the same user name and password for both local authentication and RADIUS authentication.

- To create a local network login user name and password, use the following command and specify the parameter: *user-name*

  ```
  create netlogin local-user user-name {encrypted} {password} {vlan-vsa [[{tagged | untagged} [vlan_name] | vlan_tag]]} {security-profile security_profile}
  ```

  If you attempt to create a user name with more than 32 characters, the switch displays the following messages:

  ```
  %% Invalid name detected at '^' marker.
  %% Name cannot exceed 32 characters.
  ```

  If you attempt to create a password with more than 32 characters, the switch displays the following message after you re-enter the password:

  ```
  Password cannot exceed 32 characters
  ```

  The encrypted option is used by the switch to encrypt the password. Do not use this option through the command line interface (CLI).
- After you enter a local network login user name, press **[Enter]**. The switch prompts you twice to enter the password.

The following example:

- Creates a new local network login user name.
- Creates a password associated with the local network login user name.

```
# create netlogin local-user megtest
password: <Enter the password. The switch does not display the password.>
Reenter password: <Re-enter the password. The switch does not display the password.>
```

For information about specifying the destination *VLAN*, see the section Specifying a Destination VLAN on page 1233.

Note
If you do not specify a password or the keyword **encrypted**, you are prompted for one.

Specifying a Destination VLAN

If you configure a local network login account with a destination *VLAN*, upon successful authentication, the client transitions to the permanent, destination VLAN.

You can specify the destination VLAN when you initially create the local network login account or at a later time.

Adding VLANs when Creating a Local Network Login Account

- To specify the destination *VLAN* when creating the local network login account, use the following command and specify the **vlan-vsa** option with the associated parameters:

  ```
  create netlogin local-user user-name {encrypted} {password} {vlan-vsa [[{tagged | untagged} [vlan_name] | vlan_tag]]} {security-profile security_profile}
  ```

  Where the following is true:

  - **tagged**—Specifies that the client be added as tagged.
  - **untagged**—Specifies that the client be added as untagged.
  - **vlan_name**—Specifies the name of the destination VLAN.
  - **vlan_tag**—Specifies the VLAN ID, tag, of the destination VLAN.

The following example:

- Creates a new local network login user name.
- Creates a password associated with the local network login user name.
- Adds the VLAN test1 as the destination VLAN.

The following is a sample display from this command:

```
create netlogin local-user megtest vlan-vsa "test1"
password: <Enter the password. The switch does not display the password.>
Reenter password: <Re-enter the password. The switch does not display the password.>
```

Adding VLANs at a Later Time

To specify the destination *VLAN* after you created the local network login account, use the following command:

```
configure netlogin local-user user-name {vlan-vsa [[{tagged | untagged} [vlan_name | vlan_tag]] | none]}
```

Where the following is true:

- **tagged**—Specifies that the client be added as tagged
- **untagged**—Specifies that the client be added as untagged
- **vlan_name**—Specifies the name of the destination VLAN
- **vlan_tag**—Specifies the VLAN ID, tag, of the destination VLAN
- **none**—Specifies that the VSA 211 wildcard (*) is applied, only if you do not specify tagged or untagged

The following example:

- Modifies a previously created local network login account.
- Specifies that clients are added as tagged to the VLAN.
- Adds the VLAN blue as the destination VLAN.

```
configure netlogin local-user megtest vlan-vsa tagged "blue"
```

Modifying an Existing Local Network Login Account

After you create a local network login user name and password, you can update the following attributes of that account:

- Password of the local network login account.
- Destination *VLAN* attributes including: adding clients tagged or untagged, the name of the VLAN, and the VLAN ID.

If you try modifying a local network login account that is not present on the switch, or you incorrectly enter the name of the account, output similar to the following appears:

```
  * Switch # configure netlogin local-user purplenet
^
%% Invalid input detected at '^' marker.
```

To confirm the names of the local network login accounts on your switch, use the following command:

```
show netlogin local-users
```

Updating the Local Network Login Password

Passwords are case-sensitive. Passwords must have a minimum of zero characters and a maximum of 32 characters.

1. Update the password of an existing local network login account with the following command:

   ```
   configure netlogin local-user user_name
   ```

   where *user_name* specifies the name of the existing local network login account.
2. After you enter the local network login user name, press **[Enter]**.

   The switch prompts you to enter a password.
3. At the prompt, enter the new password and press **[Enter]**.

   The switch then prompts you to reenter the password.

   If you attempt to create a password with more than 32 characters, the switch displays the following message after you re-enter the password:

   ```
   Password cannot exceed 32 characters
   ```

   The following example modifies the password for the existing local network login account megtest.The following is a sample display from this command:

   ```
   configure netlogin local-user megtest
   password: <Enter the new password. The switch does not display the password.>
   Reenter password: <Re-enter the new password. The switch does not display the
   password.>
   ```

After you complete these steps, the password has been updated.

Updating VLAN Attributes

You can add a destination *VLAN*, change the destination VLAN, or remove the destination VLAN from an existing local network login account.

To make any of these VLAN updates, use the following command:

```
configure netlogin local-user user-name {vlan-vsa [[{tagged | untagged}
[vlan_name | vlan_tag]] | none]}
```

Where the following are true:

- *user_name*—Specifies the name of the existing local network login account
- **tagged**—Specifies that the client be added as tagged
- **untagged**—Specifies that the client be added as untagged
- *vlan_name*—Specifies the name of the destination VLAN
- *vlan_tag*—Specifies the VLAN ID, tag, of the destination VLAN
- **none**—Specifies that the VSA 211 wildcard (*) is applied, only if you do not specify tagged or untagged

Displaying Local Network Login Accounts

To display a list of local network login accounts on the switch, including *VLAN* information, use the following command:

```
show netlogin local-users
```

Deleting a Local Network Login Account

To delete a local network login user name and password, use the following command:

```
delete netlogin local-user user-name
```

Note
You cannot mass delete all local users. You can only remove them one at a time by specifying each name.

802.1X Authentication

802.1X authentication methods govern interactions between the supplicant (client) and the authentication server.

The most commonly used methods are Transport Layer Security (TLS); Tunneled TLS (TTLS), which is a Funk/Certicom standards proposal; and PEAP.

TLS is the most secure of the currently available protocols, although TTLS is advertised to be as strong as TLS. Both TLS and TTLS are certificate-based and require a Public Key Infrastructure (PKI) that can issue, renew, and revoke certificates. TTLS is easier to deploy, as it requires only server certificates, by contrast with TLS, which requires client and server certificates. With TTLS, the client can use the RSA Data Security, Inc. *MD5 (Message-Digest algorithm 5)* Message-Digest Algorithm mode of user name/password authentication.

If you plan to use 802.1X authentication, refer to the documentation for your particular *RADIUS* server and 802.1X client on how to set up a PKI configuration.

Note
If you do not configure the reauthentication timer in non-policy mode, or the reauthentication or idle timer for policy mode, Dot1x clients remain active and are not removed when switching between machine authentication and user authentication (by logging out of a VM, and then logging back into the VM) or when switching between two users using the same VM.

Interoperability Requirements

For network login to operate, the user (supplicant) software and the authentication server must support common authentication methods.Not all combinations provide the appropriate functionality.

Supplicant Side

The supported 802.1X supplicants (clients) are Windows 7 and Windows 8 native clients, and Meetinghouse AEGIS.

A Windows 7 or Windows 8 802.1X supplicant can be authenticated as a computer or as a user. Computer authentication requires a certificate installed in the computer certificate store, and user authentication requires a certificate installed in the individual user's certificate store.

By default, the Windows 7 or Windows 8 machine performs computer authentication as soon as the computer is powered on, or at link-up when no user is logged into the machine. User authentication is performed at link-up when the user is logged in.

Windows 7 or Windows 8 also supports guest authentication, but this is disabled by default. Refer to relevant Microsoft documentation for further information. You can configure a Windows 7 or Windows 8 machine to perform computer authentication at link-up even if a user is logged in.

Authentication Server Side

The *RADIUS* server used for authentication must be EAP-capable. Consider the following when choosing a RADIUS server:

- Types of authentication methods supported on RADIUS, as mentioned previously.
- Need to support VSAs. Parameters such as *`Extreme-Netlogin-Vlan-Name`* (destination vlan for port movement after authentication) and *`Extreme-NetLogin-Only`* (authorization for network login only) are brought back as VSAs.

- Need to support both EAP and traditional user name-password authentication. These are used by network login and switch console login respectively.

Note

For information on how to use and configure your RADIUS server, refer to Configuring the RADIUS Client on page 1611 and to the documentation that came with your RADIUS server.

Enabling and Disabling 802.1X Network Login

Network Login must be disabled on a port before you can delete a *VLAN* that contains that port. You can set a reauthentication maximum counter value to indicate the number of reauthentication trials after which the supplicant is denied access or given limited access.

- To enable 802.1X network login on the switch, use the following command: `enable netlogin dot1x` Any combination of types of authentication can be enabled on the same switch. At least one of the authentication types must be specified on the CLI.
- To disable 802.1X network login on the switch, use the following command: `disable netlogin dot1x`
- To enable 802.1X network login on one or more ports, use the following command: `enable netlogin ports portlist dot1x`
- To disable 802.1X network login on one or more ports, use the following command: `disable netlogin ports portlist dot1x`
- To configure the reauthentication counter values, use the following command: `configure netlogin dot1x timers`

802.1X Network Login Configuration Example

The following configuration example shows the Extreme Networks switch configuration needed to support the 802.1X network login example.

Note

In the following sample configuration, any lines marked (Default) represent default settings and do not need to be explicitly configured.

```
create vlan "temp"
create vlan "corp"
configure vlan "default" delete ports 4:1-4:4
# Configuration Information for VLAN corp
# No VLAN-ID is associated with VLAN corp.
configure vlan "corp" protocol "ANY" (Default)
configure vlan "corp" ipaddress 10.203.0.224 255.255.255.0
# Configuration Information for VLAN Mgmt
configure vlan "Mgmt" ipaddress 10.10.20.30 255.255.255.0
# Network Login Configuration
configure netlogin vlan "temp"
enable netlogin dot1x
enable netlogin ports 1:10-1:14, 4:1-4:4 dot1x
```

```
# RADIUS Configuration
configure radius netlogin primary server 10.0.1.2 1812 client-ip 10.10.20.30 vr "VR-Mgmt"
configure radius netlogin primary shared-secret purple
enable radius
```

The following example is for the FreeRADIUS server; the configuration might be different for your *RADIUS* server:

```
#RADIUS Server Setting, in this example the user name is eaptest
eaptest Auth-Type := EAP, User-Password == "eaptest"
Session-Timeout = 120,
Termination-Action =1
```

For information about how to use and configure your RADIUS server, refer to Configuring the RADIUS Client on page 1611 and the documentation that came with your RADIUS server.

Configuring Guest VLANs

Ordinarily, a client that does not respond to 802.1X authentication remains disabled and cannot access the network.

802.1X authentication supports the concept of "guest *VLAN*s" that allow such a supplicant (client) limited or restricted network access. If a supplicant connected to a port does not respond to the 802.1X authentication requests from the switch, the port moves to the configured guest VLAN. A port always moves untagged into the guest VLAN.

Note
The supplicant does not move to a guest VLAN if it fails authentication after an 802.1X exchange; the supplicant moves to the guest VLAN only if it does not respond to an 802.1X authentication request.

When the authentication server sends an 802.1X request to the supplicant, there is a specified time interval for the supplicant to respond. By default, the switch uses the supplicant response timer to authenticate the supplicant every 30 seconds for a maximum of three tries. If the supplicant does not respond within the specified time, the authentication server sends another request. After the third 802.1X request without a supplicant response, the port is placed in the guest VLAN, if the guest VLAN feature has been configured for the port. The number of authentication attempts is not a user-configured parameter.

If a supplicant on a port in the guest VLAN becomes 802.1X-capable, the switch starts processing the 802.1X responses from the supplicant. If the supplicant is successfully authenticated, the port moves from the guest VLAN to the destination VLAN specified by the *RADIUS* server. If the RADIUS server does not specify a destination VLAN, the port moves to the VLAN it belonged to before it was placed in the guest VLAN. After a port has been authenticated and moved to a destination VLAN, it is periodically

re-authenticated. If the port fails authentication, it moves to the VLAN it belonged to originally.

Note
A guest VLAN is not a normal network login VLAN. A guest VLAN performs authentication only if authentication is initiated by the supplicant.

Using Guest VLANs

Suppose you have a meeting that includes company employees and visitors from outside the company.

In this scenario, your employees have 802.1X enabled supplicants but your visitors do not. By configuring a guest *VLAN*, when your employees log into the network, they are granted network access (based on their user credentials and 802.1X enabled supplicants). However, when the visitors attempt to log into the network, they are granted limited network access because they do not have 802.1X enabled supplicant. The visitors might be able to reach the Internet, but they are unable to access the corporate network.

For example, in Figure 155 Host A has 802.1x capability and Host B does not. When Host A is authenticated, it is given full access to the network. Host B does not have 802.1X capability and therefore does not respond to 802.1X requests from the switch. If port B is configured with the guest VLAN, port B is moved to the guest VLAN. Then Host B will be able to access the Internet but not the corporate network. After Host B is equipped with 802.1X capability, it can be authenticated and allowed to be part of the corporate network.

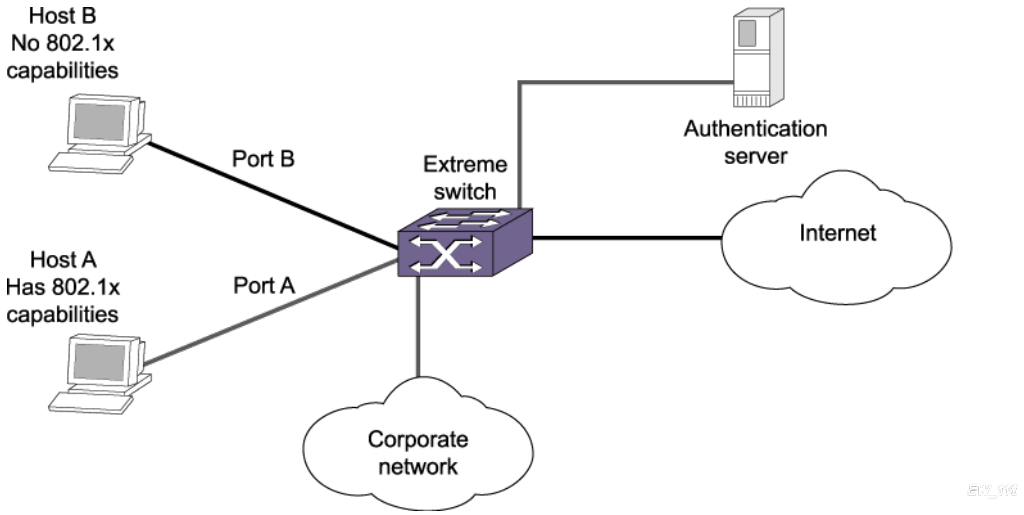


Figure 155: Guest VLAN for Network Login

Guidelines for Configuring Guest VLANs

Keep in mind the following guidelines when configuring guest *VLAN*s:

- You must create a VLAN and configure it as a guest VLAN before enabling the guest VLAN feature.
- Configure guest VLANs only on network login ports with 802.1x enabled.

- Movement to guest VLANs is not supported on network login ports with MAC-based or web-based authentication.
- 802.1x must be the only authentication method enabled on the port for movement to guest VLAN.
- No supplicant on the port has 802.1x capability.

Creating Guest VLANs

If you configure a guest *VLAN*, and a supplicant has 802.1X disabled and does not respond to 802.1X authentication requests from the switch, the supplicant moves to the guest VLAN. Upon entering the guest VLAN, the supplicant gains limited network access.

Note
You can configure guest VLANs on a per port basis, which allows you to configure more than one guest VLAN per VR. In ExtremeXOS 11.5 and earlier, you can only configure guest VLANs on a per VR basis, which allows you to configure only one guest VLAN per VR.

To create a guest VLAN, use the following command:

```
configure netlogin dot1x guest-vlan vlan_name {ports port_list}
```

Enabling Guest VLANs

To enable the guest *VLAN*, use the following command:

```
enable netlogin dot1x guest-vlan ports [all | ports]
```

Modifying the Supplicant Response Timer

The default supplicant response timeout is 30 seconds, and the range is 1-120 seconds. The number of authentication attempts is not a user-configured parameter.

To modify the supplicant response timer, use the following command and specify the **supp-resp-timeout** parameter:

```
configure netlogin dot1x timers [{server-timeout server_timeout}
{quiet-period quiet_period} {reauth-period reauth_period {reauth-max
max_num_reauths}} {supp-resp-timeout supp_resp_timeout}]
```

Disabling Guest VLANs

To disable the guest *VLAN*, use the following command:

```
disable netlogin dot1x guest-vlan ports [all | ports]
```

Unconfiguring Guest VLANs

To unconfigure the guest *VLAN*, use the following command:

```
unconfigure netlogin dot1x guest-vlan {ports port_list | vlan_name}
```

Displaying Guest VLAN Settings

To display the guest *VLAN* settings, use the following command:

```
show netlogin guest-vlan {vlan_name}
```

If you specify the *vlan_name*, the switch displays information for only that guest VLAN.

The output displays the following information in a tabular format:

- Port—Specifies the 802.1X enabled port configured for the guest VLAN.
- Guest-vlan—Displays guest VLAN name and status: enable/disable.
- Vlan—Specifies the name of the guest VLAN.

Post-authentication VLAN Movement

After the supplicant has been successfully authenticated and the port has been moved to a *VLAN*, the supplicant can move to a VLAN other than the one it was authenticated on.

This occurs when the switch receives an Access-Accept message from the *RADIUS* server with a VSA that defines a new VLAN. The supplicant remains authenticated during this transition. This occurs on both untagged and tagged VLANs. For example, suppose a supplicant submits the required credentials for network access; however, it is not running the current, approved anti-virus software or it does not have the appropriate software updates installed. If this occurs, the supplicant is authenticated but has limited network access until the problem is resolved. After you update the supplicant's anti-virus software, or install the software updates, the RADIUS server re-authenticates the supplicant by sending ACCESS-ACCEPT messages with the accompanying VLAN attributes, thereby allowing the supplicant to enter its permanent VLAN with full network access.

This is normal and expected behavior; no configuration is necessary.

802.1X Authentication and Network Access Protection

802.1X authentication in combination with Microsoft's Network Access Protection (NAP) provide additional integrity checks for end users and supplicants that attempt to access the network.

NAP allows network administrators to create system health policies to ensure supplicants that access or communicate with the network meet administrator-defined system health requirements. For example, if a supplicant has the appropriate software updates or anti-virus software installed, the supplicant is deemed healthy and granted network access. On the other hand, if a supplicant does not have the appropriate software updates or anti-virus software installed, the supplicant is deemed unhealthy and is placed in a quarantine *VLAN* until the appropriate update or anti-virus software is installed. After the supplicant is healthy, it is granted network access. For more information about NAP, please refer to the documentation that came with your Microsoft Windows or Microsoft Server software.

To configure your network for NAP, the minimum required components are:

- Extreme Networks switches running ExtremeXOS 11.6 or later.
- *RADIUS* server that supports NAP (Microsoft Windows Vista operating system refers to this as a network policy server (NPS), formerly known as the internet authentication server (IAS)).
- Remediation servers that receive unhealthy supplicants. The remediation servers contain the appropriate software updates, anti-virus software, and so on to make a supplicant healthy.

In addition to the required hardware and software, you must configure NAP-specific VSAs on your RADIUS server. By configuring these VSAs, you ensure supplicant authentication and authorization to the network and the switch creates dynamic Access Control Lists (ACLs) to move unhealthy supplicants to the quarantine VLAN for remediation. For more information see, Using NAP-Specific VSAs to Authenticate 802.1X Supplicants on page 1245.

Figure 156 displays a sample network that uses NAP to protect the network.

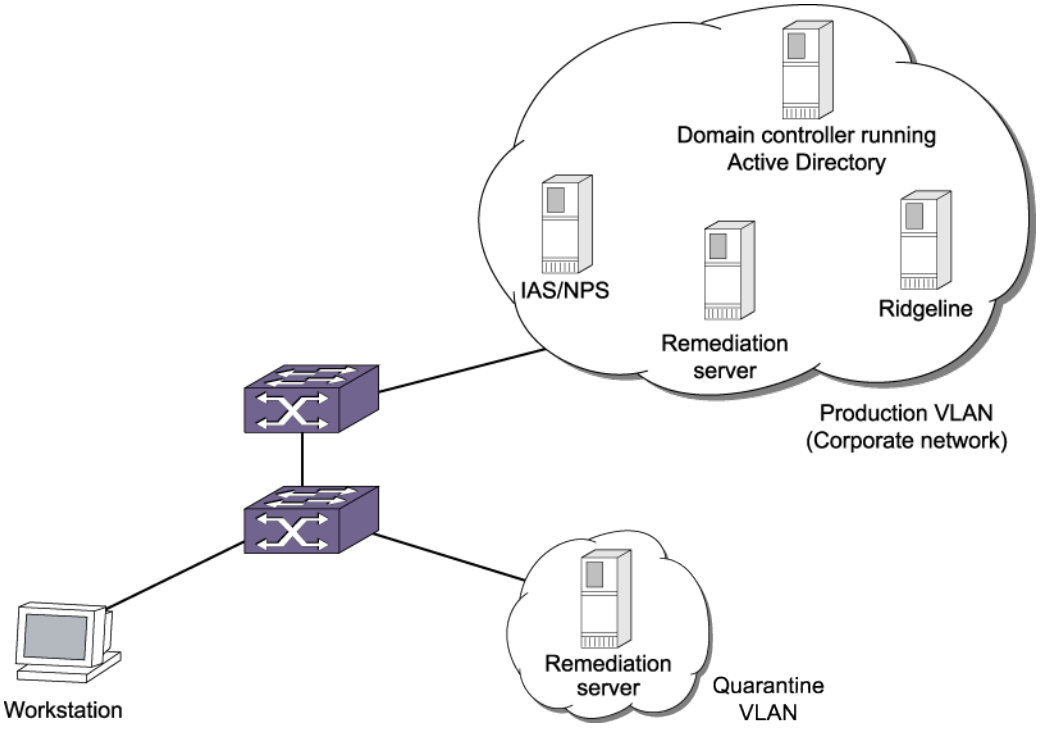


Figure 156: Sample Network Using NAP to Provide Enhanced Security

Example Scenarios Using NAP

Using the following figure, the following two scenarios describe some sample actions taken when an 802.1X-enabled supplicant initiates a connection to the network.

The scenarios assume the following:

- Scenario 1 has a healthy 802.1X-enabled supplicant.
- Scenario 2 has an unhealthy 802.1X-enabled supplicant.

- 802.1X network login has been configured and enabled on the switch.
- The *RADIUS* server has been configured using the NAP-specific VSAs for authenticating supplicants.
- The remediation servers have been configured with the appropriate software updates, anti-virus software, and so on.
- *VLAN*s Production and Quarantine have already been created and configured.

Note

You can dynamically create the quarantine VLAN if you configure dynamic VLAN creation on the switch. For more information see, Configuring Dynamic VLANs for Network Login on page 1263.

Scenario 1--Healthy Supplicant

The steps to authenticate a healthy supplicant are:

1. The 802.1X supplicant initiates a connection to the 802.1X network access server (NAS), which in this scenario is the Extreme Networks switch.
2. The supplicant passes its authentication credentials to the switch using PEAP and an inner authentication method such as MS-CHAPv2.
3. The *RADIUS* server requests a statement of health (SoH) from the supplicant.

   Only NAP-capable supplicants create an SoH, which contains information about whether or not the supplicant is compliant with the system health requirements defined by the network administrator.
4. If the SoH indicates that the supplicant is healthy, the RADIUS server sends an Access-Accept message with a RADIUS VSA indicating which *VLAN* the healthy supplicant is moved to (in this example, the Production VLAN).
5. The switch authenticates the supplicant and moves it into the Production VLAN.
6. The switch sends a trap to the NMS indicating that the supplicant has been successfully authenticated and the VLAN into which it has been moved.

Scenario 2--Unhealthy Supplicant

The steps to authenticate an unhealthy supplicant are:

1. The 802.1X supplicant initiates a connection to the 802.1X network access server (NAS), which in this scenario is the Extreme Networks switch.
2. The supplicant passes its authentication credentials to the switch using PEAP and an inner authentication method such as MS-CHAPv2.
3. The *RADIUS* server requests a statement of health (SoH) from the supplicant.

   Only NAP-capable supplicants create an SoH, which contains information about whether or not the supplicant is compliant with the system health requirements defined by the network administrator.
4. If the SoH indicates that the supplicant is unhealthy, the RADIUS server sends an Access-Accept message with RADIUS VSAs indicating which:
   - *VLAN* the unhealthy supplicant is moved to (in this example, the Quarantine VLAN).
   - the remediation server(s) from which the supplicant can get software updates, anti-virus software and so on to remediate itself.

5. When the switch receives the VLAN and remediation server information from the RADIUS server, the switch:
   - Moves the supplicant into the Quarantine VLAN.
   - Applies ACLs to ensure the supplicant in the Quarantine VLAN can access only the remediation servers
   - Drops all other traffic not originating/destined from/to the remediation servers
   - .
6. The supplicant connects to the remediation server to get software updates, anti-virus software, and so on to get healthy.
7. After the supplicant is healthy, it restarts the authentication process and is moved to the Production VLAN, as a healthy supplicant with full network access.

Using NAP-Specific VSAs to Authenticate 802.1X Supplicants

Table 81 contains the VSA definitions for 802.1X network login in conjunction with devices and servers that support NAP.

The Microsoft Vendor ID is 311.

Note
For more information about NAP and the VSAs supported by NAP, please refer to the documentation that came with your Microsoft operating system or server.

Table 81: NAP-Specific VSA Definitions for 802.1X Network Login

VSA	Vendor Type	Type	Sent-in	Description
MS-Quarantine-State	45	Integer	Access-Accept	Indicates the network access level that the *RADIUS* server authorizes the user. The network access server (the switch) also enforces the network access level. A value of “0” gives the user full network access. A value of “1” gives the user limited network access. A value of “2” gives the user full network access within a specified time period.
MS-IPv4-Remediation-Servers	52	Integer	Access-Accept	Indicates the IP address(es) of the remediation server(s) that an unhealthy supplicant moves to in order to get healthy.

ACLS for Remediation Servers

The NAP VSA, MS-IPv4-Remediation-Servers, contains a list of IP addresses that an unhealthy and therefore quarantined supplicant should be allowed access to so that it can remediate itself and become healthy.

The way a quarantine is implemented on the switch is simply by moving the client/port to a user-designated 'quarantine' *VLAN* whose VLANID/Name is sent in the Access-

Accept message. It is up to the user to ensure that the quarantine VLAN does indeed have limited access to the rest of the network. Typically, this can be done by disabling IP forwarding on that VLAN so no routed traffic can get out of that VLAN. Also, with dynamic VLAN creation, the quarantine VLAN being supplied by *RADIUS* could be dynamically created on the switch, once dynamic VLAN creation is enabled on it. The remediation server(s) would need to be accessible via the uplink port, regardless of whether the quarantine VLAN is pre-configured or dynamically created, since IP forwarding is not enabled on it.

To get around this restriction, network login has been enhanced so when a MS-Quarantine-State attribute is present in the Access-Accept message with extremeSessionStatus being either 'Quarantined' or 'On Probation,' then a 'deny all traffic' dynamic *ACL (Access Control List)* will be applied on the VLAN. If such an ACL is already present on that VLAN, then no new ACL will be applied.

When the last authenticated client has been removed from the quarantine VLAN, then the above ACL will be removed.

Additionally, if the MS-IPv4-Remediation-Servers VSA is present in the Access-Accept message, for each IP address present in the VSA a 'permit all traffic to/from this IP address' ACL will be applied on the quarantine VLAN. This will allow traffic to/from the remediation servers to pass unhindered in the Quarantine VLAN while all other traffic will be dropped.

Web-Based Authentication

This section describes web-based network login.

For web-based authentication, you need to configure the switch DNS name, default redirect page, session refresh, and logout-privilege. URL redirection requires the switch to be assigned a DNS name. The default name is network-access.com. Any DNS query coming to the switch to resolve switch DNS name in unauthenticated mode is resolved by the DNS server on the switch in terms of the interface (to which the network login port is connected).

Note
When both HTTP and HTTPS are enabled on the switch and sending HTTP requests from the Netlogin client, HTTPS takes preference and the switch responds with a HTTPS response.

Enabling and Disabling Web-Based Network Login

- To enable web-based network login on the switch, use the following command:

  ```
  enable netlogin web-based
  ```

  Any combination of types of authentication can be enabled on the same switch. At least one of the authentication types must be specified on the CLI.
- To disable web-based network login on the switch, use the following command:

  ```
  disable netlogin web-based
  ```

- To enable web-based network login on one or more ports, use the following command:

  ```
  enable netlogin ports portlist web-based
  ```

  Network Login must be disabled on a port before you can delete a *VLAN* that contains that port.
- To disable web-based network login on one or more ports, use the following command:

  ```
  disable netlogin ports portlist web-based
  ```

Configuring the Base URL

- To configure the network login base URL, use the following command:

  ```
  configure netlogin base-url url
  ```

  where *url* is the DNS name of the switch.

  For example, configure netlogin base-url network-access.com makes the switch send DNS responses back to the network login clients when a DNS query is made for network-access.com.

Configuring the Redirect Page

To configure the network login redirect page, use the following command:

```
configure netlogin redirect-page url
```

Where *url* defines the redirection information for the users after they have logged in. You must configure a complete URL starting with http:// or https://. By default, the redirect URL value is "http://www.extremenetworks.com" and default re-direction will take maximum of 20 seconds i.e default netlogin-lease-timer + 10 seconds. Re-direct time can be changed by tuning netlogin-lease-timer.

You can also configure the redirect value to a specific port number, such as 8080. For example, you can configure the network login redirect page to the URL value "http://www.extremenetworks.com:8080". The default port value is 80.

This redirection information is used only in case the redirection info is missing from *RADIUS* server. For example, configure netlogin base-url http://www.extremenetworks.com redirects all users to this URL after they get logged in.

For more information about SSH2, see Secure Shell 2 on page 1644.

Configuring Proxy Ports

To configure the ports to be hijacked and redirected, use the following command:

```
configure netlogin add proxy-port tcp_port {http | https}

configure netlogin delete proxy-port
```

For each hijacked or proxy port, you must specify whether the port is to be used for HTTP or HTTPS traffic. No more than five hijack or proxy ports are supported for HTTP in addition to port 80 (for HTTP) and port 443 (for HTTPS), both of which cannot be deleted.

Configuring Session Refresh

To enable or disable the network login session refresh, use the following commands:

```
enable netlogin session-refresh {refresh_minutes}

disable netlogin session-refresh
```

Where *minutes* ranges from 1 - 255. The default setting is 3 minutes. The commands `enable netlogin session-refresh` and `configure netlogin session-refresh` make the logout window refresh itself at every configured time interval. Session refresh is enabled by default. When you configure the network login session refresh for the logout window, ensure that the *FDB* aging timer is greater than the network login session refresh timer.

Note
If an attempt is made to authenticate the client in a non-existent *VLAN*, and the move fail action setting is authenticate, then the client is successfully authenticated in the port's original VLAN, but subsequent session refreshes fail and cause the client to become unauthenticated.

When web-based Network Login is configured with proxy ports and session-refresh are also enabled, you must configure the web browser to bypass the web proxy server for the IP address of the VLAN into which the client moves after authentication.

Configuring Logout Privilege

- To enable or disable network login logout privilege, use the following commands:

  ```
  enable netlogin logout-privilege
  disable netlogin logout-privilege
  ```

  These commands turn the privilege for network login users to logout by popping up (or not popping up) the logout window. Logout-privilege is enabled by default.
- You can configure the number of times a refresh failure is ignored before it results in the client being unauthenticated by using the following commands:

  ```
  configure netlogin allowed-refresh-failures
  unconfigure netlogin allowed-refresh-failures
  ```

  You can set the number of failures to be from between 0 and 5. The default number of logout failures is 0.

Configuring the Login Page

You can fully customize the HTML login page and also add custom embedded graphical images to it. This page and the associated graphics must be uploaded to

the switch so that they can be served up as the initial login page at the base URL. Both HTTP and HTTPS are supported as a means of authenticating the user via the custom page.

In general, the steps for setting up a custom login page and graphical images (if any) are as follows:

1. Write the custom webpage.
2. TFTP the page and any embedded JPEG or GIF graphical images that it references onto the switch.
3. Enable and configure web-based Network Login on the switch. When the custom page is present on the switch, it will override the configured banner.

Configuring a Network Login Banner

- To configure a banner on a network login screen, use the following command:
  `configure netlogin banner` *`banner`*
- To display configured banners from the network login screen, use the following command:
  `show netlogin banner`
- To clear configured network login banners, use the following command:
  `unconfigure netlogin banner`

Login Page Contents

The customized web-page must have the file name `netlogin_login_page.html`.

While the contents of the page are left up to the customer, they must contain the following elements:

- An HTML submit form with action="/hello" and method="post" that is used to send the Network Login username and password to the switch. The form must contain the following:
  - A username input field with name="extremenetloginuser"
  - A password input field with name="extremenetloginpassword"
- Optionally, one or more graphical images embedded using the tags

  `<img src="netlogin_<xxx>.jpg">` or

  `<img src="netlogin_<xxx>.jpeg">` or

  `<img src="<netlogin_<xxx>.gif">`

  where `<xxx>` is user-configurable.

The following is a sample custom page, where the embedded graphical image is named netlogin_welcome.jpg:

```
<html lang="en">

<head>
<title>Network Login Page</title>
</head>
```

```
<body>
<form action="/hello" method="post">
<img src="netlogin_welcome.jpg">
<br/>
Please log in:

<br/>
User:

<input type="text" name="extremenetloginuser" />
<br/>
Password:

<input type="password" name="extremenetloginpassword" />
<br/>
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

Uploading the Login File to the Switch

To upload the page and the JPEG/GIF files, the switch TFTP command must be used.

For example, assuming the page resides on a TFTP server with IP address 10.255.49.19, the command used would be:

```
BD-8810.1 # tftp get 10.255.49.19 netlogin_login_page.html
```

General Guidelines

The following general guidelines are applicable to the login page:

- When the custom web page is not present on the switch, the system falls back to the using the default banner. The web page may be added (or removed) from the switch at any time, at which point the switch will stop (or start) using the banner.
- The graphical image file names referenced in the web page must not have any path information prepended.
- Both uppercase and lowercase names (or a mixture) for the graphical image filenames are supported, but the user and password tag names should be either all uppercase or all lowercase, not a mixture of the two.
- More than one form may exist on the page. This can be useful when, for example, in addition to the main username and password that is typed in by the user, an additional special username and password needs to be auto-filled and sent. This could be used when end users without a valid username or password need to be given restricted access to the network.

Limitations

The following limitations apply to the login page:

- When the client is in the unauthenticated state, any embedded URLs in the custom page are inaccessible to it.
- Only JPEG and GIF graphical images are supported.

- It is the web page writer's responsibility to write the HTML page correctly and without errors.
- Only TFTP is supported as a method to upload the web-page and graphical images to the switch.

Customizable Authentication Failure Response

In the event of web-based network login authentication failure, you can use a custom authentication failure page to recover.

When a customized login page is in effect, by default, any authentication failure results in the following failure response being delivered to the browser:

```
Login Incorrect. Click here to try again.
```

Clicking on the indicated link will bring the user back to the initial custom login page.

You may choose to override the above default response with a custom one. This custom failure response page must be uploaded to the switch using TFTP with the name `netlogin_login_fail_page.html`. When authentication fails, the switch responds with this page. If the page is deleted from the switch, the response reverts back to the default.

The same graphical images that are uploaded to the switch for the custom login page can also be embedded in the custom authentication failure page.

Note
The custom authentication failure page can be used only when authentication is being done via the custom login page.

Customizable Graphical Image in Logout Popup Window

You can embed a graphical image in the logout popup window.

This image appears in the window in addition to the text that is displayed. The image must be TFTPed to the switch in the same manner as the custom login image, and must have the filename `netlogin_logout_image.jpg` or `netlogin_logout_image.gif` (depending on whether the image is JPEG or GIF). If no such image is present on the switch, then the logout popup contains only text.

Web-Based Network Login Configuration Example

The following configuration example shows both the Extreme Networks switch configuration and the *RADIUS* server entries needed to support the example.

VLAN corp is assumed to be a corporate subnet which has connections to DNS, WINS servers, network routers, and so on. VLAN temp is a temporary VLAN and is created to provide connections to unauthenticated network login clients. Unauthenticated ports belong to the VLAN temp. This kind of configuration provides better security as unauthenticated clients do not connect to the corporate subnet and will not be able to

send or receive any data. They have to get authenticated in order to have access to the network.

- ISP Mode—Network login clients connected to ports 1:10–1:14, VLAN corp, will be logged into the network in ISP mode. This is controlled by the fact that the VLAN in which they reside in unauthenticated mode and the RADIUS server Vendor Specific Attributes (VSA), Extreme-Netlogin-Vlan, are the same, corp. So there will be no port movement. Also if this VSA is missing from RADIUS server, it is assumed to be ISP Mode.
- Campus Mode—On the other hand, clients connected to ports 4:1–4:4, VLAN temp, will be logged into the network in Campus mode since the port will move to the VLAN corp after getting authenticated. A port moves back and forth from one VLAN to the other as its authentication state changes.

Both ISP and Campus mode are not tied to ports but to a user profile. In other words, if the VSA Extreme:Extreme-Netlogin-Vlan represents a VLAN different from the one in which the user currently resides, then VLAN movement will occur after login and after logout. In following example, it is assumed that campus users are connected to ports 4:1–4:4, while ISP users are logged in through ports 1:10–1:14.

Note
In the following sample configuration, any lines marked (Default) represent default settings and do not need to be explicitly configured.

```
create vlan “temp”
create vlan “corp”
configure vlan “default” delete ports 4:1-4:4
enable ipforwarding
# Configuration Information for VLAN temp
# No VLAN-ID is associated with VLAN temp.
configure vlan “temp” ipaddress 198.162.32.10 255.255.255.0
# Configuration Information for VLAN corp
# No VLAN-ID is associated with VLAN corp.
configure vlan “corp” ipaddress 10.203.0.224 255.255.255.0
configure vlan “corp” add port 1:10 untagged
configure vlan “corp” add port 1:11 untagged
configure vlan “corp” add port 1:12 untagged
configure vlan “corp” add port 1:13 untagged
configure vlan “corp” add port 1:14 untagged
# Network Login Configuration
configure vlan “temp” dhcp-address-range 198.162.32.20 - 198.162.32.80
configure vlan “temp” dhcp-options default-gateway 198.162.32.1
configure vlan “temp” dhcp-options dns-server 198.162.32.10
configure vlan “temp” dhcp-options wins-server 10.0.1.85
configure netlogin vlan “temp”
enable netlogin web-based
enable netlogin ports 1:10-1:14,4:1-4:4 web-based
configure netlogin base-url "network-access.com" (Default)
configure netlogin redirect-page http://www.extremenetworks.com (Default)
enable netlogin logout-privilege (Default)
disable netlogin session-refresh  3 (Default)
# DNS Client Configuration
configure dns-client add name-server 10.0.1.1
configure dns-client add name-server 10.0.1.85
#RADIUS Client Configuration
configure radius netlogin primary server 10.0.1.2 1812 client-ip 10.10.20.30 vr “Vr-Mgmt”
configure radius netlogin primary shared-secret purple
enable radius
```

For this example, the following lines (for a FreeRADIUS server) should be added to the RADIUS server users file for each user:

```
Extreme:Extreme-Netlogin-Only = Enabled (if no CLI authorization)
Extreme:Extreme-Netlogin-Vlan = "corp" (destination vlan for CAMPUS mode network login)
```

Note
For information about how to use and configure your RADIUS server, refer to Configuring the RADIUS Client on page 1611 and the documentation that came with your RADIUS server.

Web-Based Authentication User Login

To use web-based authentication:

1. Set up the Windows IP configuration for DHCP.
2. Plug into the port that has web-based network login enabled.
3. Log in to Windows.
4. Release any old IP settings and renew the DHCP lease.

   This is done differently depending on the version of Windows the user is running:

   - **Windows 9x**—Use the winipcfg tool. Choose the Ethernet adapter that is connected to the port on which network login is enabled. Use the buttons to release the IP configuration and renew the DHCP lease.
   - **Windows 7 or Windows 8**—Use the ipconfig command line utility. Use the command ipconfig/release to release the IP configuration and ipconfig/renew to get the temporary IP address from the switch. If you have more than one Ethernet adapter, specify the adapter by using a number for the adapter following the `ipconfig` command. You can find the adapter number using the command `ipconfig/all`. At this point, the client will have its temporary IP address. In this example, the client should have obtained an IP address in the range 198.162.32.20–198.162.32.80.

     Note
     The idea of explicit release/renew is required to bring the network login client machine in the same subnet as the connected *VLAN*. When using web-based authentication, this requirement is mandatory after every logout and before login again as the port moves back and forth between the temporary and permanent VLANs.

5. Bring up the browser and enter any URL as `http://www.123.net or http://1.2.3.4` or switch IP address as `http://<IP address>/login` (where IP address could be either temporary or Permanent VLAN Interface for Campus mode).

   URL redirection redirects any URL and IP address to the network login page. This is significant where security matters most, as no knowledge of VLAN interfaces is required to be provided to network login users, because they can login using a URL or IP address.

   Note
   URL redirection requires that the switch be configured with a DNS client.

   A page opens with a link for Network Login.
6. Click the Network Login link.

   A dialog box opens requesting a user name and password.
7. Enter the user name and password configured on the *RADIUS* server. After the user has successfully logged in, the user will be redirected to the URL configured on the RADIUS server. During the user login process, the following takes place:
   a. Authentication is done through the RADIUS server.
   b. After successful authentication, the connection information configured on the RADIUS server is returned to the switch:
      - The permanent VLAN
      - The URL to be redirected to (optional)
      - The URL description (optional)
   c. The port is moved to the permanent VLAN.
   d. You can verify this using the `show vlan` command. For more information on the `show vlan` command, see Displaying VLAN Information on page 2229.

After a successful login has been achieved, there are several ways that a port can return to a non-authenticated, non-forwarding state:

- The user successfully logs out using the logout web browser window.
- The link from the user to the switch's port is lost.
- There is no activity on the port for 20 minutes.
- An administrator changes the port state.

Note
Because network login is sensitive to state changes during the authentication process, we recommend that you do not log out until the login process is complete. The login process is complete when you receive a permanent address.

MAC-Based Authentication

MAC-based authentication is used for supplicants that do not support a network login mode, or supplicants that are not aware of the existence of such security measure (for example, an IP phone).

If a MAC address is detected on a MAC-based enabled network login port, an authentication request is sent once to the AAA application. AAA tries to authenticate the MAC address against the configured *RADIUS* server and its configured parameters (timeout, retries, and so on) or the local database.

In a MAC-based authentication environment, the authentication verification is done only after at MAC address detection. However, forced re-authentication is allowed through the Session-Timeout VSA when supplied along with Termination-Action attribute “RADIUS-Request” by RADIUS. When this VSA with Termination-Action attribute “RADIUS-Request” is present the switch, re-authenticates the client based on the value supplied by the VSA. If no VSA is present, there is no re-authentication. If Session-Timeout VSA is supplied and no Termination-Action attribute or “Default” Termination-Action attribute is specified, the client is unauthenticated. You can also force re-authentication by configuring the MAC re-authentication timers using the CLI (see Configuring Reauthentication Period on page 1258). If MAC re-authentication timers are configured using the CLI and RADIUS sends a different session timeout value, the RADIUS session timeout has higher precedence.

The credentials used for this are the MAC address of the supplicant in ASCII representation, and a locally configured password on the switch. If no password is configured, the MAC address is used as the password. ExtremeXOS 30.3 provides an option to send the MAC address in either uppercase or lowercase for the user name or password to provide compatibility with the requirements of the RADIUS server (see Configuring MAC Authentication Case Option for User Name and Password on page 1258).You can also group MAC addresses together using a mask.

You can configure a MAC list or a table of MAC entries to filter and authenticate clients based on their MAC addresses. If a match is found in the table of MAC entries, authentication occurs. If no match is found in the table of MAC entries, and a default entry exists, the default will be used to authenticate the client. All entries in the list are automatically sorted in longest prefix order. All passwords are stored and showed encrypted.

You can associate a MAC address with one or more ports. By learning a MAC address, the port confirms the supplicant before sending an authorization request to the RADIUS server. This additional step protects your network against unauthorized supplicants because the port accepts only authorization requests from the MAC address learned on that port. The port blocks all other requests that do not have a matching entry.

Note
When ONEPolicy is enabled and authentication required mode is configured with a static macsource rule applied, even if a MAC address fails authentication, traffic is forwarded.

Note
With ONEPolicy enabled, admin-profile port rule configured, and authentication required mode set, traffic is not forwarded by the admin profile VLAN when MAC authentication fails.

Version 33.4.1 adds support for the Message-Authenticator attribute in RADIUS Access-Requests. This enhancement aligns with security advisory SA-2024-084 and ensures the attribute is included in requests for MAC authentication, CLI-based management authentication, and Chalet-based management authentication.

Note
When a RADIUS Access-Accept message is received with a Termination-Action attribute value of RADIUS-Request and a Session-Timeout attribute value of zero, reauthentication occurs immediately, as specified in Section 3.17 of RFC 3580. These RADIUS server configuration settings can cause continuous reauthentication of 802.1X clients.

Limitations

Web-based authentication is not supported on Extended Edge Switching extended ports.

Enabling and Disabling MAC-Based Network Login

Network Login must be disabled on a port before you can delete a *VLAN* that contains that port.

- To enable MAC-based network login on the switch, use the following commands:
  `configure netlogin vlan vlan_name`

  `enable netlogin mac`

  Any combination of types of authentication can be enabled on the same switch. At least one of the authentication types must be specified on the CLI.

  If **`enable netlogin mac`** is entered, **`configure netlogin add mac-list default`** configuration is added by default. This added mac-list config will not be removed if **`disable netlogin mac`** is entered. Instead, **`configure netlogin delete mac-list default`** must be entered to remove the mac-list config.
- To disable MAC-based network login on the switch, use the following command:
  `disable netlogin mac`
- To enable MAC-based network login on one or more ports, use the following command:
  `enable netlogin ports portlist mac`
- To disable MAC-based network login on one or more ports, use the following command:
  `disable netlogin ports portlist mac`

Associating a MAC Address to a Specific Port

You can configure the switch to accept and authenticate a client with a specific MAC address. Only MAC addresses that have a match for the specific ports are sent for authentication. For example, if you associate a MAC address with one or more ports, only authentication requests for that MAC address received on the port(s) are

sent to the configured *RADIUS* server or local database. The port(s) block all other authentication requests that do not have a matching entry. This is also known as secure MAC.

- To associate a MAC address with one or more ports, specify the **ports** option when using the following command:

  ```
  configure netlogin add mac-list [mac {mask} | default] {encrypted}
  {password} {ports port_list}
  ```

  You must enable MAC-based network login on the switch and the specified ports.

  If MAC-based network login is not enabled on the specified port(s), the switch displays a warning message similar to the following:

  ```
  WARNING: Not all specified ports have MAC-Based NetLogin enabled.
  ```

For a sample configuration, see Securing MAC Configuration Example on page 1258.

Adding and Deleting MAC Addresses

- To add a MAC address to the table, use the following command:

  ```
  configure netlogin add mac-list [mac {mask} | default] {encrypted}
  {password} {ports port_list}
  ```

- To remove a MAC address from the table, use the following command:

  ```
  configure netlogin delete mac-list [mac {mask} | default]
  ```

Displaying the MAC Address List

To display the MAC address table, use the following command:

```
show netlogin mac-list
```

When a client needs authentication the best match will be used to authenticate to the server.

MAC-based authentication is VR aware, so there is one MAC list per VR.

Assume we have a supplicant with MAC address 00:04:96:05:40:00, and the switch has the following table:

```
MAC Address/Mask        Password (encrypted)      Port(s)
--------------------  ----------------------  --------------
00:00:00:00:00:10/48    <not configured>          1:1-1:5
00:00:00:00:00:11/48    <not configured>          1:6-1:10
00:00:00:00:00:12/48    <not configured>          any
00:01:30:70:0C:00/48    yaqu                      any
00:01:30:32:7D:00/48    ravdqsr                   any
00:04:96:00:00:00/24    <not configured>          any
```

The user name used to authenticate against the *RADIUS* server would be “000496000000,” as this is the supplicant's MAC address with the configured mask applied. Although this is the default, ExtremeXOS 16.1 allows for a hyphenated mac address to be sent - `configure netlogin mac username format hyphenated`.

Note that the commands are VR aware, and therefore one MAC list table exists per VR.

Configuring MAC Authentication Case Option for User Name and Password

If a RADIUS server is case-sensitive and it expects the user name and password only in lowercase, then MAC Authentication will fail due to the case mismatch, since whenever a MAC address is used as user name and password, by default the MAC address is sent only in uppercase.

ExtremeXOS 30.3 provides an option to send the Network Login (NetLogin) MAC Authentication MAC address in either upper or lower case for the user name or password.

To change the MAC case, use the following command:

```
configure netlogin mac username case {lower | upper}
```

Configuring Reauthentication Period

The reauthentication timer is applicable for all the cases when session timeout is not sent by *RADIUS*. Clients moved to authentication success, authentication failure VLAN, and service unavailable VLANs can be reauthenticated using this timer. If the switch receives the session-timeout attribute during authentication, the switch uses that value to set the reauthentication period.

To configure the reauthentication period for network login MAC-based authentication, use the following command:

```
configure netlogin mac timers reauth-period
```

For more information about RADIUS server attributes, see Configuring the RADIUS Client on page 1611.

Securing MAC Configuration Example

The following configuration example shows how to configure secure MAC on your Extreme Networks switch. To configure secure MAC:

1. Create a *VLAN* used for network login.
2. Configure the VLAN for network login.
3. Enable MAC-based network login on the switch.
4. Enable MAC-based network login on the ports used for authentication.
5. Specify one or more ports to accept authentication requests from a specific MAC address.

   In the following example, authentication requests from MAC address:

   - 00:00:00:00:00:10 are only accepted on ports 1:1 through 1:5
   - 00:00:00:00:00:11 are only accepted on ports 1:6 through 1:10

- 00:00:00:00:00:12 are accepted on all other ports

```
create vlan nlvlan
configure netlogin vlan nlvlan
enable netlogin mac
enable netlogin ports 1:1-1:10 mac
configure netlogin add mac-list 00:00:00:00:00:10 ports 1:1-1:5
configure netlogin add mac-list 00:00:00:00:00:11 ports 1:6-1:10
configure netlogin add mac-list 00:00:00:00:00:12
```

To view your network login configuration, use the following commands:

```
show netlogin {port portlist vlan vlan_name} {dot1x {detail}} {mac}
{web-based}

show netlogin mac-list
```

MAC-Based Network Login Configuration Example

The following configuration example shows the Extreme Networks switch configuration needed to support the MAC-based network login example.

```
create vlan "temp"
create vlan "corp"
configure vlan "default" delete ports 4:1-4:4
# Configuration Information for VLAN corp
# No VLAN-ID is associated with VLAN corp.
configure vlan "corp" ipaddress 10.203.0.224 255.255.255.0
# Network Login Configuration
configure netlogin vlan "temp"
enable netlogin mac
enable netlogin ports 4:1-4:4 mac
configure netlogin add mac-list default <password>
# RADIUS Client Configuration
configure radius netlogin primary server 10.0.1.2 1812 client-ip 10.10.20.30 vr "VR-Mgmt"
configure radius netlogin primary shared-secret purple
enable radius
```

The following example is a user's file entry for a specific MAC address on a FreeRADIUS server:

```
00E018A8C540 Auth-Type := Local,  User-Password == "00E018A8C540"
```

Note

For information about how to use and configure your *RADIUS* server, refer to Configuring the RADIUS Client on page 1611 and the documentation that came with your RADIUS server.

MAC-Based Authentication Delay

Prior to ExtremeXOS 21.1, the default behavior was to authenticate clients using all enabled authentication methods on a port—MAC, 802.1X, and Web-based—for backward compatibility. By default, MAC authentication was triggered immediately upon receiving the first packet from the client.

To delay MAC authentication, users can configure the MAC authentication delay period using the CLI. The default value is 0 seconds, but it can be set between 0 and 120 seconds.

Example scenario

Assume MAC, 802.1X, and Web-based authentication are enabled on a port. When a client connects:

- With no delay configured, ExtremeXOS immediately performs MAC authentication, applies the associated action, and then proceeds with 802.1X authentication if triggered.
- If 802.1X is preferred over MAC, the MAC authentication action is unapplied, and the 802.1X action is applied.
- Only one authentication method (MAC or 802.1X) can use the **Extreme-URL-Redirect VSA**. The URL from the first authenticated method is processed and used; changes in authentication order do not affect this.

To delay or bypass MAC authentication, configure the MAC authentication delay period per port. When the first packet is received, ExtremeXOS waits for the configured delay period before triggering MAC authentication, allowing time for preferred methods like 802.1X to initiate.

In this case, the user "Adam" can authenticate using 802.1X during the delay period. The time ExtremeXOS waits before initiating MAC authentication is the MAC authentication delay period, and it is fully user-configurable.

Additional Network Login Configuration Details

This section describes additional, optional network login configurations.These configurations are not required to run network login; however, depending on your network settings and environment, you can use the commands described in this section to enhance your network login settings.

Review the earlier sections of this chapter for general information about network login and information about MAC-based, web-based, and 802.1X authentication methods.

Configuring Network Login MAC-Based VLANs

Currently, network login allows only a single, untagged *VLAN* to exist on a port. This limits the flexibility for untagged supplicants because they must be in the same VLAN.

ExtremeSwitching series switches support network login MAC-based VLANs. Network login MAC-based VLANs allow a port assigned to a VLAN to operate in a MAC-based fashion. This means that each individual untagged supplicant, identified by its MAC address, can be in different VLANs.

Network login MAC-based VLAN utilizes VSA information from both the network login local database and the *RADIUS* server. After successfully performing the Campus mode operation, the supplicant is added untagged to the destination VLAN.

To support this feature, you must configure the network login port's mode of operation.

Network Login MAC-Based VLANs Rules and Restrictions

This section summarizes the rules and restrictions for configuring network login MAC-based VLANss:

- You must configure and enable network login on the switch and before you configure network login MAC-based VLANs.

  If you attempt to configure the port's mode of operation before enabling network login, the switch displays an error message similar to the following:

  ```
  ERROR: The following ports do not have NetLogin enabled; 1
  ```

- If you attempt to configure network login MAC-based VLANs on 10 Gigabit Ethernet ports, the switch displays an error message similar to the following:

  ```
  ERROR: The following ports do not support the MAC-Based VLAN mode; 1, 2, 10
  ```

  In ExtremeXOS version 12.0 and later, you can configure MAC-based-VLANs on 10 Gigabit Ethernet ports.
- You can have a maximum of 1,024 MAC addresses per switch.

Configuring the Port Mode

To support network login MAC-based VLANs on a network login port, you must configure that port's mode of operation.

Specify MAC-based operation using the following command and specifying mac-based-vlans:

```
configure netlogin ports [all | port_list] mode [mac-based-vlans | port-based-vlans]
```

By default, the network login port's mode of operation is port-based-vlans. If you modify the mode of operation to mac-based-vlans and later disable all network login protocols on that port, the mode of operation automatically returns to port-based-vlans.

When you change the network login port's mode of operation, the switch deletes all currently known supplicants from the port and restores all VLANs associated with that port to their original state. In addition, by selecting mac-based-vlans, you are unable to manually add or delete untagged VLANs from this port. Network login now controls these VLANs.

With network login MAC-based operation, every authenticated client has an additional *FDB* flag that indicates a translation MAC address. If the supplicant's requested VLAN does not exist on the port, the switch adds the requested VLAN.

Configuring Ingress Filtering

Ingress filtering is disabled by default when MAC-based VLANs are enabled on the port. To toggle the ingress filtering setting for MAC-based VLANs, use the following command:

- `configure ports port_list` **`mac-based-vlans ingress-filtering`** `[`**`on`**`|`**`off`**`]`

  The configuration is specific to MAC-based VLANs, and only takes effect if MAC-based VLANs are enabled on the port. If the command is executed on a port that does not have MAC-based VLANs enabled, the command will be accepted, but it will not be applied until MAC-based VLANs are enabled.

Displaying Network Login MAC-Based VLAN Information

The following commands display important information for network login MAC-based *VLAN*s.

FDB Information

To view *FDB* entries, use the following command:

```
show fdb netlogin [all | mac-based-vlans]
```

By specifying netlogin, you see only FDB entries related to network login or network login MAC-based *VLAN*s.

The flags associated with network login include:

- v—Indicates the FDB entry was added because the port is part of a MAC-Based virtual port/VLAN combination
- n—Indicates the FDB entry was added by network login

VLAN and Port Information

- To view the *VLAN*s that network login adds temporarily in MAC-based mode, use the following command:
  `show ports port_list information detail`

  By specifying **`information`** and **`detail`**, the output displays the temporarily added VLANs in network login MAC-based mode.

  This command will also show port information for ingress filtering for MAC-based VLANs.
- To confirm this, review the following output of this command:
  VLAN cfg—The term "MAC-based" appears next to the tag number.

  Netlogin port mode—This output was added to display the port mode of operation. "Mac-based" appears and the network login port mode of operation.
- To view information about the ports that are temporarily added in MAC-based mode for network login, due to discovered MAC addresses, use the following command:
  `show vlan detail`

  By specifying **`detail`**, the output displays detailed information including the ports associated with the VLAN.

The flags associated with network login include:

- a—Indicates that egress traffic is allowed for network login
- u—Indicates that egress traffic is not allowed for network login.
- m—Indicates that the network login port operates in MAC-based mode.

Note
If network login is enabled together with *STP*, the 'a' and 'u' flags are controlled by network login only when the STP port state is 'Forwarding.'

Network Login MAC-Based VLAN Example

The following example configures the network login MAC-based *VLAN* feature:

```
create vlan users12
create vlan nlvlan
configure netlogin vlan nlvlan
enable netlogin mac
enable netlogin ports 1:1-1:10 mac
configure netlogin ports 1:1-1:10 mode mac-based-vlans
configure netlogin add mac-list default MySecretPassword
```

Expanding upon the previous example, you can also utilize the local database for authentication rather than the *RADIUS* server:

```
create netlogin local-user 000000000012 vlan-vsa untagged default
create netlogin local-user 000000000010 vlan-vsa untagged users12
```

For more information about local database authentication, see Local Database Authentication on page 1232.

Configuring Dynamic VLANs for Network Login

During an authentication request, network login receives a destination *VLAN* (if configured on the *RADIUS* server) to put the authenticated user in.

The VLAN must exist on the switch for network login to authenticate the client on that VLAN.

You can configure the switch to dynamically create a VLAN after receiving an authentication response from a RADIUS server. A dynamically created VLAN is only a Layer 2 bridging mechanism; this VLAN does not work with routing protocols to forward traffic. If configured for dynamic VLAN creation, the switch automatically creates a supplicant VLAN that contains both the supplicant's physical port and one or more uplink ports. After the switch unauthenticates all of the supplicants from the dynamically created VLAN, the switch deletes that VLAN.

Note
Dynamically created VLANs do not support the session refresh feature of web-based network login because dynamically created VLANs do not have an IP address.

By dynamically creating and deleting VLANs, you minimize the number of active VLANs configured on your edge switches. In addition, the dynamic VLAN name can

be stored on the RADIUS server and supplied to the switch during authentication, simplifying switch management. A key difference between dynamically created VLANs and other VLANs is that the switch does not save dynamically created VLANs. Even if you use the save command, the switch does not save a dynamically created VLAN.

After you configure network login on the switch, the two steps to configure dynamic VLANs are:

- Specifying the tagged uplink port(s) to be added to each dynamically created VLAN.
- Enabling the switch to create dynamic VLANs.

Specifying the Uplink Ports

The uplink ports send traffic to and from the supplicants from the core of the network. Uplink ports should not be configured for network login (network login is disabled on uplink ports).

To specify one or more ports as tagged uplink ports that are added to the dynamically created *VLAN*, use the following command:

```
configure netlogin dynamic-vlan uplink-ports [port_list | none]
```

By default, the setting is none.

If you specify an uplink port with network login enabled, the configuration fails and the switch displays an error message similar to the following:

```
ERROR: The following ports have NetLogin enabled: 1, 2
```

If this occurs, select a port with network login disabled.

Enabling Dynamic VLANs for Network Login

By default, the setting is disabled.

To enable the switch to create dynamic *VLAN*s, use the following command:

```
configure netlogin dynamic-vlan [disable | enable]
```

When enabled, the switch dynamically creates VLANs. Remember, dynamically created VLANs are not permanent nor are user-created VLANs. The switch uses the VLAN ID supplied by the *RADIUS* attributes (as described below) to create the VLAN. The switch only creates a dynamic VLAN if the requested VLAN, indicated by the VLAN ID, does not currently exist on the switch.

The RADIUS server uses VSAs to forward VLAN information.The forwarded information can include only a VLAN ID (no VLAN name). The following list specifies the supported VSAs for configuring dynamic VLANs:

- Extreme: Netlogin-VLAN-ID (VSA 209)
- Extreme: Netlogin-Extended-VLAN (VSA 211)
- IETF: Tunnel-Private-Group-ID (VSA 81)

Extreme-specific VSAs have greater precedence than IETF Tunnel-Private-Group-Id attribute if both are sent using RADIUS Access-Accept. For example, if Extreme-

Netlogin-Extended-Vlan 211: Uv1;Tcorp and Tunnel-Private-Group-Id:200 are both sent, MAC is authenticated on VLAN v1 and corp, and not on vlanid 200.

Note
If the ASCII string contains only numbers, it is interpreted as the VLAN ID. Dynamic VLANS support only numerical VLAN IDs; VLAN names are not supported.

The switch automatically generates the VLAN name in the following format: SYS_VLAN_TAG where *TAG* specifies the VLAN ID. For example, a dynamic VLAN with an ID of 10 has the name `SYS_VLAN_0010`.

Note
Like all VLAN names, dynamic VLAN names are unique. If you create a VLAN and use the name of an existing dynamic VLAN, the switch now sees the dynamic VLAN as a user-created VLAN and will save this VLAN to the switch configuration. If this occurs, the switch does not delete the VLAN after the supplicants are authenticated and moved to the permanent VLAN.

For more information on Extreme Networks VSAs, see Extreme Networks VSAs on page 1619.

Dynamic VLAN Example with Web-Based Network Login

After you finish the web-based network login configuration as described in Web-Based Network Login Configuration Example on page 1251, complete the dynamic *VLAN* configuration by:

- Assigning one or more non-network-login ports as uplink ports

  Note
  Do not enable network login on uplink ports. If you specify an uplink port with network login enabled, the configuration fails and the switch displays an error message.

- Enabling the switch to dynamically create VLANs

Whether you have MAC-based, web-based, or 802.1X authentication, you use the same two commands to configure dynamic VLANs on the switch.

The following example configures dynamic VLANs on the switch:

```
configure netlogin dynamic-vlan uplink ports 2:1-2:2
configure netlogin dynamic-vlan enable
```

Displaying Dynamic VLAN Information

- To display summary information about all of the *VLAN*s on the switch, including any dynamically VLANs currently operating on the switch, use the following command:

  ```
  show vlan
  ```

  If the switch has dynamically created VLANs, the VLAN name begins with `SYS_NLD_`.

- To display the status of dynamic VLAN configuration on the switch, use the following command:

  ```
  show netlogin
  ```

  The switch displays the current state of dynamic VLAN creation (enabled or disabled) and the uplink port(s) associated with the dynamic VLAN.

Configuring Network Login Port Restart

You can configure network login to restart specific network-login-enabled ports when the last authenticated supplicant unauthenticates, regardless of the configured authentication methods on the port.

This feature, known as *network login port restart* , is available with all network login authentication methods although is most practical with web-based network login. This section describes how this feature behaves with web-based network login; MAC-based and 802.1X network login do not experience any differences in behavior if you enable network login port restart.

Currently with web-based network login, if you have an authenticated supplicant and log out of the network, you must manually release the IP address allocated to you by the *DHCP* server. The DHCP server dynamically manages and allocates IP addresses to supplicants. When a supplicant accesses the network, the DHCP server provides an IP address to that supplicant. DHCP cannot renegotiate their leases, which is why you must manually release the IP address.

For example, if the idle timer expires on the switch, the switch disconnects your network session. If this occurs, it may be unclear why you are unable to access the network. After you manually renew the IP address, you are redirected to the network login login page and can log back into the network. To solve this situation in a single supplicant per port environment, port restart triggers the DHCP client on the PC to restart the DHCP address assignment process.

Guidelines for Using Network Login Port Restart

Configure network login port restart on ports with directly attached supplicants.

If you use a hub to connect multiple supplicants, only the last unauthenticated supplicant causes the port to restart. Although the hub does not inflict harm to your network, in this situation, the previously unauthenticated supplicants do not get the benefit of the port restart configuration.

Enabling Network Login Port Restart

To enable network login port restart, use the following command:

```
configure netlogin ports [all | port_list] restart
```

Disabling Network Login Port Restart

To disable network login port restart, use the following command:

```
configure netlogin [ports port_list | all] allow egress-traffic
{all_cast | broadcast | none | unicast}

configure netlogin ports [all | port_list] no-restart
```

Displaying the Port Restart Configuration

To display the network login settings on the port, including the configuration for port restart, use the following command:

```
show netlogin port port_list
```

Output from this command includes the enable/disable state for network login port restart.

Authentication Failure and Services Unavailable Handling

ExtremeXOS provides the following features for handling network login authentication failures, and for handling instances of services unavailable:

- Configure Authentication Failure VLAN
- Configure Authentication Services Unavailable VLAN
- Configure Reauthentication Period (`configure netlogin mac ports [port_list | all] timers [{reauth-period [reauth_period]} {reauthentication [on| off]} {delay [delay_period]}]`)

You can use these features to set and control the response to network login authentication failure and instances of services unavailable.

Configuring Authentication Failure VLAN

When a network login client fails authentication, it is moved to authentication failure *VLAN* and given restricted access.

To configure the authentication failure VLAN, use the following commands:

```
configure netlogin authentication failure vlan

unconfigure netlogin authentication failure vlan

enable netlogin authentication failure vlan ports

disable netlogin authentication failure vlan ports
```

Use the command **`enable netlogin authentication failure vlan`** to configure authentication failure VLAN on network-login-enabled ports. When a supplicant fails authentication, it is moved to the authentication failure VLAN and is given limited access until it passes the authentication.

Through either a *RADIUS* or local server, the other database is used to authenticate the client depending on the authentication database order for that particular network login method (mac, web, or dot1x). If the final result is authentication failure and if the

authentication failure VLAN is configured and enabled on that port, then the client is moved there.

For example, if the network login MAC authentication database order is local, radius and the authentication of a MAC client fails through local database, then the RADIUS server is used to authenticate. If the RADIUS server also fails authentication, the client is moved to the authentication failure VLAN. This applies for all authentication database orders (radius,local; local,radius; radius; local).

In the above example if authentication through local fails but passes through the RADIUS server, the client is moved to appropriate destination VLAN. If the local server authentication fails and the RADIUS server is not available, the client is not moved to authentication failure VLAN.

Only when Radius Access-Reject is received, the client moves to auth fail vlan in Netlogin Mac. 802.1x Authentication: Client will move to auth fail vlan for access rejection and for cases where Radius Access Challenge is received without reject from the Radius Server. After the expiration of the timeout period, client will move to auth fail vlan.

Dependency on authentication database order

There are four different authentication orders which can be configured per authentication method. These four orders are the following:

- RADIUS
- Local
- RADIUS, Local
- Local, RADIUS

For each authentication order, the end result is considered in deciding whether to authenticate the client through the authentication failure *VLAN* or the authentication service unavailable VLAN (if configured).

For example, if the authentication order is radius, local, with the RADIUS server unavailable, and local authentication failed, the client is authenticated in the authentication failure VLAN (if one is configured on the port).

For local authentication, the following cases are considered as authentication failure.

- If the user is not created in the local database.
- If the user is configured, but the password does not match.

If the user is configured, but the password does not match, it is considered an authentication failure.

For RADIUS server authentication, if for some reason the user cannot be authenticated due to problems with the RADIUS configuration, the RADIUS server not running, or some other problem then it is considered as an authentication service unavailable. If the actual authentication fails then it is considered as an authentication failure.

Configuring Authentication Service-Unavailable VLAN

When the authentication service is not available for authentication, the supplicant is moved to authentication service-unavailable *VLAN* and given restricted access.

Starting with ExtremeXOS 30.2, you can configure multiple (10 maximum) service-unavailable VLANs per port.

To configure the authentication services unavailable VLAN, use the following commands:

```
configure netlogin authentication service-unavailable [{add} | {delete}
| {{vlan vlan_name} {ports port_list {tagged | untagged}}}]

unconfigure netlogin authentication service-unavailable vlan vlan_name
{ports port_list}

enable netlogin authentication service-unavailable vlan ports [ports |
all]

disable netlogin authentication service-unavailable vlan ports [ports |
all]
```

If a NetLogin port has web enabled, authentication-failure VLAN and authentication service-unavailable VLAN configuration are not applicable to MAC and Dot1x clients connected to that port. For example, if port 1:2 has NetLogin MAC and web authentication enabled and the service-unavailable VLAN is configured and enabled on it, and if a MAC client connected to that port fails authentication, it is not moved to service-unavailable VLAN.

Fallback VLANs in Policy

In version 33.2.1, the fallback VLAN functionality provided with Netlogin with Policy disabled can also be achieved with Policy enabled by using Policy admin rules. A policy profile is assigned to a port (or port/mac address combination). The profile has a default pvid specified and pvid-status enabled. Authentication mode "optional" must be enabled for both Netlogin MAC and dot1x enabled ports in order to handle first authentication failure due to service unavailability. If authentication fails or the RADIUS server is unavailable, the user will be placed in the VLAN specified as the profile's default pvid (the **show netlogin session** command will display the entries as "Auth Failed".

Windows dot1x clients must be set with "Fallback to unauthorized network access". In this case, because the authentication has failed, the session is marked as such but the traffic is handled by the admin profile as long as the authentication mode is set as "optional".

An enhancement is added to version 33.2.1 that applies to existing Netlogin sessions that fail to re-authenticate because of service unavailable. This enhancement works with both authentication modes: "optional" and "required." However, "optional" is required for the admin-profile to allow traffic to flow. The user can specify an option to retain the Netlogin session, including the original VLAN assignment in this case.

This feature only works on re-authentication, because the unavailability or failure of the first authentication is handled by the existing administration rule Policy feature.

Configuration Examples

You can active this enhancement by enabling the following command options:

`configure netlogin mac authentication database-order radius` (only required for MAC-auth and not dot1x)

`configure netlogin keep-session-reauth-svc-unavail` **`on`**

`configure radius netlogin keep-alive interval 78`

`configure radius netlogin keep-alive` **`on`**

Limitations

The following are limitations for this enhancement:

- Netlogin dot1x session-timeout value sent using RADIUS must be higher than the radius keep-alive timer value.
- "Keep Session On Re-authentication Service Unavailable - On/Off" attribute doesn't apply for non-Policy Netlogin service unavailability feature.
- For dot1x, this feature depends on RADIUS keep-alive being enabled (on).
- It is recommended that **`keep-session-on-reauth-svc-unavail`** remain off for non-Policy mode.
- For re-authentication during service unavailable, it is possible that the dot1x client is triggered to re-authenticate due to timing of the re-authentication period and RADIUS keep-alive are synchronized. It is recommended that the dot1x client be set to use "Fallback to unauthorized network access". When this happens, the dot1x re-authentication timer does not get re-initialized but the session is still kept.

Supported Platforms

All platforms.

Netlogin Policy Session Re-authentication

Version 33.2.1 adds support for enabling OnePolicy to have a fallback VLAN in case the authentication service is not available when re-authentication occurs. This allows the VLAN assignment by the policy to be kept. See the previous entry on Fallback VLANs.

Supported Platforms

All platforms.

New CLI Command

The following new command supports this feature:

`configure netlogin keep-session-reauth-svc-unavail`

RADIUS Netlogin Keep Alive

Version 33.2.1 adds support for a keep alive message on the Netlogin RADIUS server. This checks for RADIUS server reachability in order to limit unnecessary re-authentication.

Supported Platforms

All platforms.

New CLI Commands:

The following new commands support this feature:

```
configure radius netlogin keep-alive

configure radius netlogin keep-alive interval
```

Modified CLI Command

The **`show radius`** command is updated with Netlogin RADIUS server information (lines 20-21):

```
# show radius
RADIUS Default State:   disabled
RADIUS Default Timeout: 3 seconds
RADIUS Algorithm: standard
RADIUS Retries: 3
RADIUS port bounce VSA: off
RADIUS dynamic-authorization: disabled
RADIUS TLS TCP Timeout: default
RADIUS TLS OCSP: off
RADIUS TLS OCSP Attributes:
  Nonce               : off
  Signer ocsp-nocheck : off
  Override Server URL : none
Switch Management RADIUS: disabled *
Switch Management RADIUS server connect time out: 3 seconds *
Switch Management RADIUS Accounting: disabled *
Switch Management RADIUS Accounting server connect time out: 3 seconds
Netlogin RADIUS: enabled
Netlogin RADIUS server connect time out: 3 seconds *
Netlogin RADIUS server keep alive: on
Netlogin RADIUS server keep alive interval: 60 seconds
Netlogin RADIUS server's last known reachability status: unreachable
Netlogin RADIUS Accounting: disabled *
Netlogin RADIUS Accounting server connect time out: 3 seconds

Primary Netlogin RADIUS server: Status is Active
    host name     :
    IP address    :  1.2.3.4
    Server IP Port:  1812
    Protocol      :  UDP
    Client address:  10.127.2.33 (VR-Mgmt)
    Retries       :  3 *
    Timeout       :  3 *
    shared secret :  #$S4/Mj804b1yid8IYgRCNF2qdlGKD+A==
Access Requests   :  157        Access Accepts    :  0
Access Rejects    :  0          Access Challenges :  0
Access Retransmits:  117        Client timeouts   :  156
Bad authenticators:  0          Unknown types     :  0
Round Trip Time   :  0
```

```
Legend: An asterisk (*) indicates a global value is in use.
```

NEW! Automate RADIUS Reauthentication Pause Using Passive Server Reachability Detection

Version 33.7.1 adds automated, passive detection of RADIUS server availability to trigger and clear the RADIUS reauthentication pause state, eliminating the need for manual CLI intervention or UPM scripts during a RADIUS outage.

Enhancement: Previously, the RADIUS reauthentication pause had to be triggered manually using a run-time CLI command. This enhancement allows the switch to automatically enter the pause state when a RADIUS timeout or failure is passively detected — without introducing active polling traffic to the network. While in the pause state, reauthentication timers for authenticated sessions are extended by a configurable maximum time, and new sessions can be assigned to a configured fallback VLAN. When RADIUS availability is passively restored through organic authentication traffic, the switch automatically resumes reauthentication timers.

Key Capabilities:

- Automatic pause state entry on passive detection of RADIUS server failure — no manual CLI command or UPM script required
- Configurable maximum pause duration to bound the outage window
- New sessions directed to a configured fallback VLAN during the pause period
- Automatic resumption of reauthentication timers when RADIUS availability is passively confirmed
- Manual override commands available to start or stop the pause state independently of automatic detection

CLI Commands

You can use the following commands to enable or disable automatic pause state entry on passive RADIUS failure detection:

```
configure netlogin radius reauthentication pause
```

```
configure netlogin radius reauthentication pause auto-start [enable | disable]
```

Manually start the RADIUS reauthentication pause:

```
start netlogin radius reauthentication pause
```

Manually stop the RADIUS reauthentication pause:

```
stop netlogin radius reauthentication pause
```

Limitations

Because this feature uses passive detection only, detecting that the RADIUS server has come back online relies on organic authentication requests from clients. If no

new clients attempt to authenticate, the switch remains in the paused state until the configured maximum timeout expires.

Supported Platforms

All Universal platforms.

NEW! Automatic Session Recovery After RADIUS Fallback VLAN Pause

Version 33.7.1 adds automatic re-authentication for sessions that were moved to the RADIUS fallback VLAN when the reauthentication pause was first triggered, restoring full network access without manual intervention after the RADIUS server becomes available again.

Enhancement: When the RADIUS reauthentication pause is activated, the initial session that detected the timeout is placed into the fallback VLAN to maintain basic connectivity. Previously, when the RADIUS server returned and the pause was lifted, that session remained in the fallback state until an administrator intervened manually. This enhancement introduces an automated re-trigger mechanism: the switch maintains a dynamic recovery list of MAC and 802.1X entries that were moved to the fallback VLAN as the pause trigger. When the pause ends — either by timer expiration or by a successful RADIUS response for another client — the switch automatically initiates re-authentication for all sessions on the recovery list.

Key Capabilities:

- Dynamic recovery list tracks sessions placed in the fallback VLAN at pause trigger time, across multiple ports simultaneously
- On pause removal, 802.1X sessions receive a port restart; MAC-Auth sessions receive a targeted re-authentication request
- Zero manual intervention required to restore full network access after a RADIUS outage

You can use the following command to configure the automatic session recovery behavior:

```
configure netlogin radius reauthentication resume
```

Limitations

For 802.1X sessions, successful recovery depends on the client endpoint correctly responding to the port reset initiated by the switch.

Supported Platforms

All Universal platforms.

RADIUS Reauthentication Pause with Passive Polling

Version 33.6.1 adds passive polling capability to RADIUS reauthentication pause, enabling automatic early resume when RADIUS service becomes available during scheduled maintenance windows.

Reauthentication Pause Overview

When RADIUS servers require scheduled maintenance, reauthentication pause allows you to temporarily suspend RADIUS reauthentication for successfully authenticated sessions. This prevents session disconnections during the maintenance window while maintaining security through the existing authentication state.

Key characteristics of reauthentication pause:

- Works only with NetLogin policy mode (not legacy mode)
- Applies to MAC and 802.1X authentication (web authentication not supported)
- Affects only sessions in authenticated success state
- New client authentication attempts may still occur if the RADIUS server hasn't been shut down yet
- Pause is time-bounded with automatic resume after the configured period
- Does not persist across reboots (temporary operational state)

Passive Polling Enhancement

Passive polling monitors RADIUS responses (Authentication Success or Authentication Failure) to detect when the service becomes available again during a pause period. Once a configurable threshold of consecutive responses is observed, reauthentication automatically resumes without requiring manual intervention.

Passive polling provides:

- Automatic early resume based on RADIUS service availability detection
- Configurable observation parameters to prevent false positives
- Conservative resume behavior using non-sliding observation windows
- Reduced manual intervention for maintenance window management

Configuring Passive Polling

Passive polling is configured separately from the pause operation itself. Configure passive polling parameters before initiating a pause, or adjust them during an active pause.

Enable or Disable Passive Polling:

```
configure netlogin radius reauthentication pause passive-poll [on | off]
```

Configure Passive Polling Parameters:

```
configure netlogin radius reauthentication pause passive-poll start-delay <minutes>
configure netlogin radius reauthentication pause passive-poll stop-period <minutes>
configure netlogin radius reauthentication pause passive-poll observe-period <minutes>
configure netlogin radius reauthentication pause passive-poll observe-threshold <count>
```

Parameters:

- **start-delay:** Time after pause begins before passive polling starts monitoring (10-1440 minutes, default 1440)
- **stop-period:** Time before scheduled resume when polling stops (10-1440 minutes, default 1440)

- **observe-period:** Window for counting RADIUS responses (5-1440 minutes, default 5)
- **observe-threshold:** Number of consecutive responses required to trigger early resume (5-512, default 512)

Important
Default values are configured to effectively disable passive polling even when turned on. You must explicitly configure appropriate values for start-delay, stop-period, and observe-threshold to enable functional passive polling.

Understanding Passive Polling Operation

Passive polling operates in phases during a reauthentication pause:

Phase 1: Delay Phase (Start Delay):

After a pause is issued, passive polling waits for the configured start-delay before beginning to monitor responses. During this time, RADIUS service is assumed unavailable. This delay allows time for the RADIUS server to be properly shut down after the pause is initiated.

Phase 2: Active Monitoring:

Passive polling monitors RADIUS responses and begins counting when the first response (Authentication Success or Authentication Failure) is received. This indicates the RADIUS service may be coming back online.

Phase 3: Observation Window:

Once a response is received, an observation window starts. If the configured observe-threshold number of consecutive responses is received within the observe-period, reauthentication automatically resumes early.

The observation period uses a non-sliding window:

- If a RADIUS timeout occurs during observation, the count resets to zero and the observation window cancels
- If the observe-threshold is not met within the observe-period, observation stops and restarts only upon receiving a new response
- This conservative approach prevents false positives from sporadic responses

Phase 4: Stop Phase (Stop Period):

Before the scheduled resume time, passive polling stops monitoring for the configured stop-period duration. This prevents unnecessary monitoring when the scheduled resume is imminent.

Timing diagram:

```
Pause      Start Delay    Observation Period           Stop Period    Scheduled
Issued     (waiting)      (active monitoring)          (stopped)      Resume
  |------------|=================~~~~~=================|------------|
  |            |                                       |            |
RADIUS assumed  Polling starts                    Polling stops   RADIUS assumed
unavailable     monitoring                                        available
```

Initiating a Reauthentication Pause

To pause RADIUS reauthentication:

```
configure netlogin radius reauthentication pause <duration>
```

Where *duration* is the pause duration in minutes (10-1440).

Example pausing for 2 hours (120 minutes):

```
configure netlogin radius reauthentication pause 120
```

During the pause:

- Sessions in authenticated success state do not reauthenticate
- Policy port authentication mode is operationally set to optional (configuration remains as-is)
- Session timeouts are affected (sessions don't timeout during pause)
- Idle timeouts continue to function normally (inactive sessions are still purged)
- Passive polling monitors for service availability if configured

Manually Resuming Reauthentication

To manually end the pause and resume reauthentication:

```
configure netlogin radius reauthentication resume
```

Manual resume is useful when:

- Maintenance completes earlier than expected
- You need to verify RADIUS server functionality before the scheduled resume
- Passive polling hasn't triggered early resume but the server is ready
- You want to end the pause for any reason

When reauthentication resumes (either automatically via passive polling, manually, or at the scheduled time):

- All paused sessions are triggered to reauthenticate
- Reauthentication timers restart at the resume point
- This may cause a burst of reauthentication requests to the RADIUS server
- AAA retries handle any drops due to the burst

Viewing Pause and Polling Status

To view the current status of reauthentication pause and passive polling:

```
show netlogin
```

Sample output when pause is not in effect:

```
NetLogin Authentication Mode : web-based DISABLED; 802.1x ENABLED; MAC-based ENABLED
NetLogin Client Aging Time : 15 minutes
Authentication Protocol Order: mac-based, 802.1x, web-based, cep
Maximum Number Of Users : 9216 (Policy Enabled only)

-----------------------------------------------
        Policy Enabled Configuration
```

```
-----------------------------------------------
Keep Session On Reauth Service Unavailable    : Off
RADIUS Reauthentication Pause                 : Not in effect
RADIUS Reauthentication Pause Passive Polling : Off
            Start delay                       : 1440 mins
            Stop period                       : 1440 mins
            Observation period                : 5 mins
            Observation threshold             : 512
            State                             : count=0 state=INVALID
```

Sample output during active pause with passive polling:

```
-----------------------------------------------
        Policy Enabled Configuration
-----------------------------------------------
Keep Session On Reauth Service Unavailable    : Off
RADIUS Reauthentication Pause                 : In effect - Duration: 120 minutes,
                                                 Time left: 3540 seconds
RADIUS Reauthentication Pause Passive Polling : On
            Start delay                       : 10 mins
            Stop period                       : 15 mins
            Observation period                : 5 mins
            Observation threshold             : 10
            State                             : count=3 state=OBSERVE (180 seconds left)
```

Passive polling state values:

- **INVALID:** No pause is active
- **START:** In start delay phase
- **ACTIVE:** Monitoring for responses but none received yet
- **OBSERVE:** Observation window is active, counting responses
- **STOP:** In stop period phase

The count field shows the current number of consecutive responses received during the observation window.

Recommended Workflow for Scheduled Maintenance

Follow this workflow for RADIUS server scheduled maintenance with passive polling:

1. **Configure passive polling parameters** (if not already configured):

   ```
   # Allow 10 minutes for server shutdown after pause
   configure netlogin radius reauthentication pause passive-poll start-delay 10

   # Stop monitoring 15 minutes before scheduled resume
   configure netlogin radius reauthentication pause passive-poll stop-period 15

   # Require 10 consecutive responses
   configure netlogin radius reauthentication pause passive-poll observe-threshold 10

   # Within a 5-minute window
   configure netlogin radius reauthentication pause passive-poll observe-period 5

   # Enable passive polling
   configure netlogin radius reauthentication pause passive-poll on
   ```

2. **Issue reauthentication pause** with duration matching your maintenance window:

   ```
   # For a 2-hour maintenance window
   configure netlogin radius reauthentication pause 120
   ```

3. **Bring down RADIUS server** for maintenance. Do this shortly after issuing the pause to ensure most sessions are captured before server shutdown.
4. **Perform server maintenance**, including testing after maintenance is complete.
5. **Bring up RADIUS server** and verify it's operational.
6. **Wait for passive polling to detect availability**:
   - Passive polling waits for start-delay, then begins monitoring
   - When responses are detected, observation window begins
   - After observe-threshold responses within observe-period, reauthentication automatically resumes

   Alternatively, **manually resume reauthentication** if you don't want to wait for passive polling:

   ```
   configure netlogin radius reauthentication resume
   ```
7. **Verify session status** after resume. Sessions that were in failed authentication state (for example, users in fallback VLAN) will age out and attempt authentication again based on aging timers.

Configuration Examples

Example 1: 2-Hour Maintenance with Aggressive Passive Polling

Scenario: Short maintenance window, want to resume as soon as possible after server is available.

```
# Configure aggressive passive polling
configure netlogin radius reauthentication pause passive-poll start-delay 10
configure netlogin radius reauthentication pause passive-poll stop-period 10
configure netlogin radius reauthentication pause passive-poll observe-threshold 5
configure netlogin radius reauthentication pause passive-poll observe-period 5
configure netlogin radius reauthentication pause passive-poll on

# Start 2-hour pause
configure netlogin radius reauthentication pause 120
```

Example 2: 4-Hour Maintenance with Conservative Passive Polling

Scenario: Longer maintenance window, want to be more certain server is stable before resuming.

```
# Configure conservative passive polling
configure netlogin radius reauthentication pause passive-poll start-delay 30
configure netlogin radius reauthentication pause passive-poll stop-period 30
configure netlogin radius reauthentication pause passive-poll observe-threshold 20
configure netlogin radius reauthentication pause passive-poll observe-period 10
configure netlogin radius reauthentication pause passive-poll on

# Start 4-hour pause
configure netlogin radius reauthentication pause 240
```

Example 3: Manual Resume Only (No Passive Polling)

Scenario: Want complete control over when to resume, prefer manual verification.

```
# Disable passive polling
configure netlogin radius reauthentication pause passive-poll off
```

```
# Start pause
configure netlogin radius reauthentication pause 180

# After maintenance is complete and verified, manually resume
configure netlogin radius reauthentication resume
```

Fallback VLANs and Admin Profile Rules

During a reauthentication pause, new clients attempting first authentication may fail if the RADIUS server is down. These clients can be directed to a fallback VLAN using policy admin profile rules.

Configuring a Blackhole VLAN for Failed Authentication:

```
# Create a VLAN with L2 forwarding disabled
create vlan blackhole_vlan tag 100
disable l2forwarding blackhole_vlan

# Create policy profile for the blackhole VLAN
configure policy profile 2 name "blackhole" pvid-status "enable" pvid 100

# Apply as admin profile rule
configure policy rule admin-profile port 1 mask 16 port-string 1 admin-pid 2
```

Configuring a DMZ VLAN for Limited Access:

Instead of completely blocking traffic, you can direct failed authentications to a DMZ VLAN with limited network access:

```
# Create DMZ VLAN with limited features
create vlan dmz_vlan tag 200
# Configure limited access on this VLAN (ACLs, routing, etc.)

# Create policy profile for DMZ
configure policy profile 3 name "dmz" pvid-status "enable" pvid 200

# Apply as admin profile rule
configure policy rule admin-profile port 1 mask 16 port-string 1 admin-pid 3
```

Timing Consideration for DMZ VLAN:

When reauthentication resumes, the authentication mode may reset to "required" if that was the configured mode before the pause. This clears FDB entries associated with the fallback VLAN. For blackhole VLANs this is not an issue, but for DMZ VLANs it causes a brief service interruption.

To minimize delay:

- Set NetLogin aging time to minimum (1 minute):
  ```
  configure netlogin aging-time 1
  ```
- Wait for sessions in DMZ VLAN to age out before resuming reauthentication
- Alternatively, manually clear sessions:
  ```
  clear netlogin state mac-address <mac>
  ```

Stack Behavior

In stack configurations:

- Only the primary switch performs the timing function for pause and passive polling
- Pause state is checkpointed to the backup; timer values are not checkpointed and are displayed as N/A
- On failover, the new primary restarts the timer from time 0
- Passive polling counter also restarts from 0 on failover
- Users can manually resume if restarting from time 0 is too long

Best Practices

- **Test in lab:** Test your passive polling configuration in a lab environment before using in production
- **Document parameters:** Document your passive polling parameter choices and the reasoning behind them
- **Conservative thresholds:** Start with conservative observe-threshold values (higher numbers) and adjust based on experience
- **Monitor during maintenance:** Watch **`show netlogin`** output during maintenance to observe passive polling behavior
- **Plan for burst:** Understand that resume causes a burst of reauthentication requests; ensure your RADIUS server can handle the load
- **Use appropriate pause duration:** Set pause duration longer than expected maintenance to allow buffer time
- **Wait for DMZ sessions:** If using DMZ VLANs, wait for failed sessions to age out before resuming
- **Have manual resume ready:** Be prepared to manually resume if passive polling doesn't trigger as expected

Troubleshooting

Passive Polling Doesn't Resume Early:

- Verify passive polling is enabled: **`show netlogin`**
- Check start-delay hasn't prevented monitoring from starting yet
- Verify observe-threshold isn't set too high for the number of authentication attempts occurring
- Confirm RADIUS server is actually responding to authentication requests
- Check if observation windows are completing successfully or resetting due to timeouts

Sessions Disconnect During Pause:

- Verify pause was issued before RADIUS server was shut down
- Check that session timeouts are longer than pause duration
- Ensure idle timeout isn't causing disconnections (idle timeout still functions during pause)
- Verify policy mode is enabled (pause only works in policy mode)

Unable to Resume Manually:

- Verify pause is actually in effect: **`show netlogin`**
- Check for NetLogin process issues
- Review system logs for error messages

Burst Overwhelms RADIUS Server:

- Ensure RADIUS server is fully operational before resuming
- Consider staggering session expirations by adjusting session timeout values
- Review RADIUS server capacity and AAA retry settings
- If needed, manually clear and reauthenticate sessions in smaller batches

Related Configuration

The pause feature works in conjunction with other NetLogin settings:

- **Keep Session on Reauth Service Unavailable:** Related but separate feature that keeps sessions when RADIUS service is unintentionally unavailable (not scheduled maintenance)
- **RADIUS Keep-Alive:** Used by 802.1X to determine RADIUS reachability and delay reauthentication when service is unavailable
- **Session Timeout:** Must be configured larger than the pause duration
- **Idle Timeout:** Continues to function during pause to remove inactive sessions
- **NetLogin Aging Time:** Affects how quickly failed sessions (in fallback VLANs) retry authentication after resume

See the "Failsafe Authentication Mode" section for configuration details on keep-session-on-reauth-svc-unavail and related settings.

NAC MAC List Command Enhancements

Version 33.5.1 adds support for NAC MAC List configuration commands incremental port management, allowing administrators to add or remove individual ports without replacing the entire port list configuration. Administrators can add or delete specific ports from existing MAC list configurations without overwriting the entire port list. This eliminates the need to reconfigure entire port lists when making small changes.

Use the following command to incrementally add or delete ports from mac-list.:

```
configure netlogin mac-list [mac {mask} | default][[add |
replace][ports update_port_list{password{encryptedencrypted_password |
password} | {encryptedencrypted_password | password}}] | [delete
portsdelete_port_list]]
```

Example

The following command replaces ports 20-25:

```
configure netlogin mac-list default replace ports 20-25
```

The following command removes ports 14-15:

```
configure netlogin mac-list default remove ports 14-15
```

The following error message will display if the given portlist does not exist:

```
conf netlogin mac-list default remove ports 10-11
Error: One or more ports in the given portlist does not exist in the configuration.
```

OAM

CFM on page 1283
Y.1731—Compliant Performance Monitoring on page 1296
Y.1731 MIB Support on page 1305
EFM OAM—Unidirectional Link Fault Management on page 1306
Two-Way Active Measurement Protocol on page 1308
Bidirectional Forwarding Detection (BFD) on page 1309

Operation, Administration, and Maintenance (OAM) includes functions used to detect network faults, measure network performance and distribute fault-related information, including CFM, Y.1731, EFM, and BFD.

CFM

Connectivity Fault Management (CFM), discussed in the IEEE 802.1Q-2011 standard and originally specified in the IEEE 802.1ag-2007 standard, allows you to detect, verify, and isolate connectivity failures in virtual bridged LANs.

Part of this specification is a toolset to manually check connectivity, which is sometimes referred to as Layer 2 ping.

Note
The ExtremeXOS implementation of CFM is based on the IEEE 802.1Q-2011 and Y.1731 standards.

There is no direct interaction between CFM and other Layer 2 protocols; however, blocked *STP (Spanning Tree Protocol)* ports are taken into consideration when forwarding CFM messages.

CFM Overview

You can create hierarchical networks, or domains, and test connectivity within a domain by sending Layer 2 messages, known as *CCM (Connectivity Check Message)*.

Note
Extreme Networks uses values defined in IEEE 802.1Q-2011 for the MAC addresses and Ethernet type for CFM.

Figure 157 shows an example of hierarchical CFM domains.

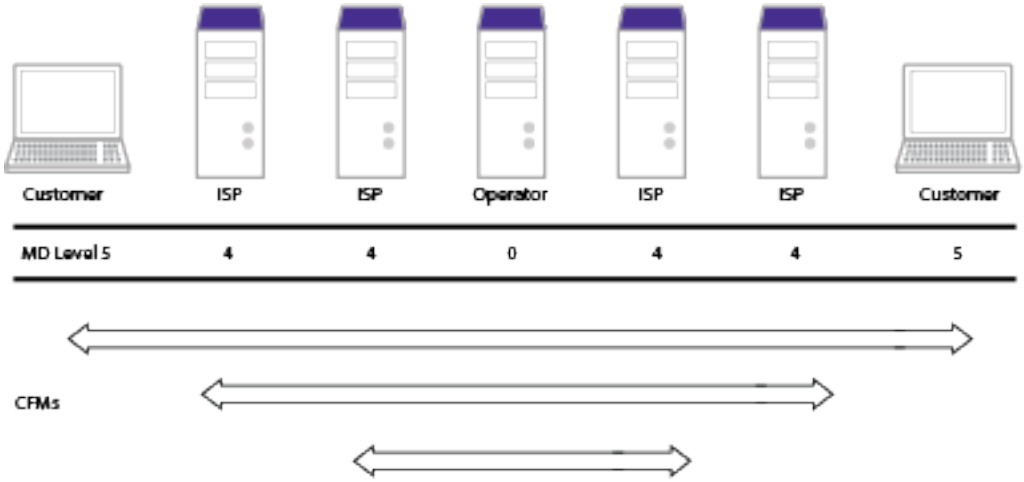


Figure 157: CFM Hierarchical Domains Example

Note
The arrows in the above figure indicate the span that CCM messages take, not the direction. (See Figure 158 on page 1285 for more information on spans for CCM messages.) This has been removed until the missing xref can be fixed.

To achieve this hierarchical connectivity testing, create and configure the following entities:

- Maintenance domains, or domains
- Maintenance domain (MD) level; a unique hierarchical numeric value for each domain
- Maintenance associations (MAs)
- Maintenance points (MPs) and *MEP (Maintenance End Point)*, which are one of the following types:
  - UP MEPs
  - DOWN MEPs
- Maintenance intermediate points (MIPs)

Note
The CFM filter function (CFF) is no longer supported from ExtremeXOS 12.1. The functionality of CFF is implicitly performed by MEPs.

An UP MEP sends CFM frames toward the frame filtering entity, which forwards the frames to all other ports of a service instance other than the port on which the UP MEP is configured. This is similar to how the frame filtering entity forwards a normal data frame, taking into account the port's *STP* state. For an UP MEP, a CFM frame exits from a port if only if the STP state of the port is in the forwarding state.

A DOWN MEP sends CFM frames directly to the physical medium without considering the port STP state. For a DOWN MEP, a CFM frame exits from a port even if the port STP state is in blocking state.

Figure 158 shows the concept of UP and DOWN MEP at logical level:

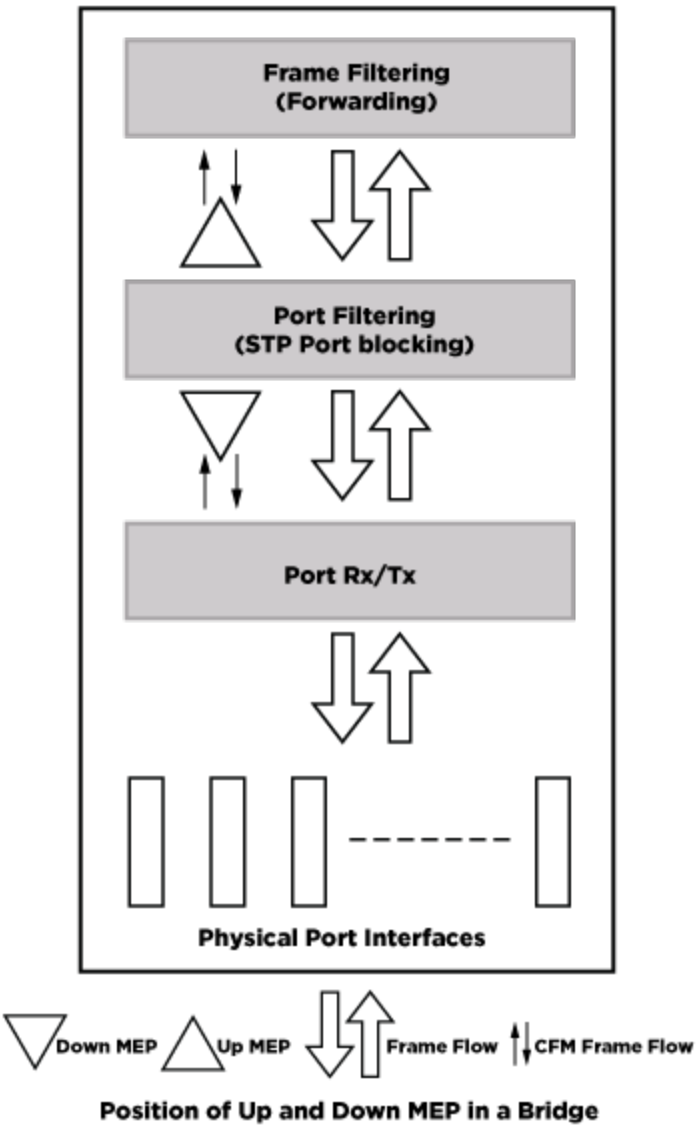


Figure 158: CFM UP and DOWN MEP at the Logical Level

You must have at least one MP on an intermediate switch in your domain. Ensure that you map and configure all ports in your domain carefully, especially the UP MEPs and the DOWN MEPs. If these are incorrectly configured, the CCMs are sent in the wrong direction in your network, and you will not be able to test the connectivity within the domain.

You can have up to eight domains on an Extreme Networks switch. A domain is the network or part of the network for which faults are to be managed; it is that

section where you are monitoring Layer 2 connectivity. A domain is intended to be fully connected internally.

Note
Domains may cross VR boundaries; domains are not -aware.

You assign each domain an MD level, which functions in a hierarchy for forwarding CFM messages. The MD levels are from 0 to 7. The highest number is superior in the CFM hierarchy.

The IEEE standard 802.1Q-2011 recommends assigning different MD levels to different domains for different network users, as follows:

- 5 to 7 for end users
- 3 and 4 for Internet service providers (ISPs)
- 0 to 3 for operators (entities carrying the information for the ISPs)

All CFM messages with a superior MD level (numerically higher) pass throughout domains with an inferior MD level (numerically lower). CFM messages with an inferior MD level are not forwarded to domains with a superior MD level. Refer to the following table for an illustration of domains with hierarchical MD levels.

Table 82: MD Levels and Recommended Use

MD level	0	1	2	3	4	5	6	7
Use	Operator			Service provider		Customer		
Superiority	Most inferior	< ----- Inferior / Superior ----- >						Most superior

Within a given domain, you create maintenance associations (MAs). Extreme Networks' implementation of CFM associates MAs with service instances (a service instance can be a *VLAN (Virtual LAN)*, VMAN, BVLAN, or SVLAN). All of the ports in that VLAN service instance are now in that MA and its associated domain. In general, you should configure one MIP on each intermediate switch in the domain and a MEP on every edge switch.

Each MA associates with one service instance, and a service instance can be associated with more than one MA. The MA is unique within the domain. One switch can have 8 domains, 128 ports, and 256 associations (see Supported Instances for CFM on page 1288).

Note
You cannot associate the Management VLAN with an MA or a domain.

You assign the MPs to ports: UP MEPs, DOWN MEPs, and MIPs. These various MPs filter or forward the CFM messages to test the connectivity of your network.

Each configured MEP periodically sends out a Layer 2 multicast or unicast CCM message.

The destination MAC address in the CCM frame is from a multicast MAC address range that is reserved for CFM messages. Each MEP must have a MEP ID that is unique within the MA. The MEPs send the CCM messages differently, depending on the configuration, as follows:

- The DOWN MEPs sends out a single CCM message.
- The UP MEPs potentially sends the CCM message to all ports on the service instance (MA)—except the sending port—depending on the MPs configured on the outgoing ports.

Note
Ensure that you configured the UP and DOWN MEPs correctly, or the CCMs will flow in the wrong direction through the domain and not allow connectivity testing.

MIPs define intermediate points within a domain. MIPs relay the CCM messages to the next MIP or MEP in the domain.

You configure the time interval for each MEP to send a CCM. We recommend setting this interval for at least one second. Each MEP also makes a note of what port and what time it received a CCM. This information is stored in the CCM database.

Each CCM has a time-to-live (TTL) value also noted for that message. This TTL interval is 3.5 times the CCM transmission interval you configured on the switch that is originating the CCM. After the TTL expires, the connectivity is considered broken, and the system sends a message to the log. One important result of the continual transmission of CCM frames is that the MAC address of the originating MEP is known to all MPs in the association.

Note
All MEPs in an MA must be configured with the same CCM transmission interval.

The MD values are from 0 to 7; in the hierarchy, the MD level of 0 is lowest and 7 is highest.

Not all combinations of MPs are allowed on the same port within an MA; only the following combinations can be on the same port within an MA:

- UP MEP and MIP
- DOWN MEP with neither UP MEP nor MIP

CFM protocol imposes the following MP restrictions within an MA on a switch:

- MA can have either up MEP or down MEP and not both.
- MA can have multiple Down MEPs.
- Only one Up MEP per MA.
- MA can have both up MEP and MIP.
- MA cannot have MIP if down MEP is present.
- Down MEPs on regular ports are created in hardware for all CCM intervals for platforms that support CFM in hardware (ExtremeSwitching X460-G2).

- Up MEPs and MEPs on *LAG (Link Aggregation Group)* ports are created in software with CCM intervals 100 msec–600000 sec on all platforms.
- Dynamic Remote MEP learning is not supported for the MEPs created in hardware. You must explicitly create static Remote MEPs.
- Sender-Id-IP Address cannot be configured for the MEPs created in hardware.
- Unicast CCM transmission is not supported by the MEPs created in hardware.
- Domain name format should be of string type to create any MEPs in hardware in that domain.
- The CCM transmission state is disabled by default for the MEPs created in hardware by the CFM user interface.
- The CCM transmission state is enabled by default for the MEPs created in hardware by CFM clients like *ERPS (Ethernet Ring Protection Switching)*.
- The hardware Remote MEP status appears in `show cfm detail`. It is also forwarded to the client if created by a client like ERPS.
- CFM objects like domain, association, MEP, Remote MEP created by a client are not saved by dot1ag.

Note
An MA can have an UP MEP in one switch and a DOWN MEP on another switch for the same MA.

Ping and Traceroute

When operators see a connectivity fault message from CCM in the system log, they can send a loopback message (LBM) or a link trace message (LTM).

These are also referred to as a Layer 2 ping or a traceroute message. You can send with an LBM or an LTM only from an MEP (either UP or DOWN).

You can only send a ping from a MEP, and you ping to the unique system MAC address on the switch you are testing connectivity to. The operator sends out a unicast LBM, and the first MIP or MEP on the switch with the destination MAC address matching the embedded MAC address replies with an LBR (loopback reply).

You can only send a traceroute (LTM) from a MEP. You send the traceroute to the unique system MAC address on the switch to which you are testing connectivity. The system sends out an LTM to the special multicast address. If the destination address is not present in the *FDB (forwarding database)*, the LTM is flooded on all the ports in the MIP node. Each MIP in the path passes the frame only in the direction of the path and sends a link trace reply (LTR) message back to the originating with information that the LTM passed. The traceroute command displays every MIP along the path (see `traceroute mac port`).

Supported Instances for CFM

Table 83 displays the CFM support in ExtremeXOS.

Table 83: ExtremeXOS CFM Support

Item	Limit	Notes
Domains	8	Per switch; one for each MD level
Associations (MAs)	256	Per switch
UP MEPs	32	Per switch
DOWN MEPs	256 hardware-placed MEPs 32	Per switch
MIPs	32	Per switch
Total CFM ports	128	Per switch; total number of all ports for all service instances assigned to an MA (see commands for ports configured for CFM)
Entries in the CCM database	2000	Number of remote end points stored in a CCM database on each MEP; 64 end points per MEP (additional CCMs discarded after this limit is reached)
CFM Segments	1000	Maximum number of CFM segments for Y.1731.
CFM Groups	1000	Maximum number of CFM groups.

Note

The total number of CFM ports is a guideline, not a limit enforced by the system.

CFM Groups

Loop detection protocols like *EAPS (Extreme Automatic Protection Switching)*/*ERPS* want to depend upon CFM to detect link status for faster failover recovery.

They register with LMEP and RMEP objects created by CFM in order to receive the link status event notifications to take the necessary action.

Currently LMEP is identified with domain, association, port, MEPId quadruples. And RMEP is identified with domain, association, LMEP, RMEPId quadruples. Each LMEP can be tied up to multiple RMEPs. So applications need to configure domain, association, LMEP and RMEPs through a client/server interface.

To simplify this, CFM provides a simple API to client applications to register/deregister CFM with a specified string name. The string name can be identified as a CFM group that binds an LMEP to multiple RMEPs. The group name is unique across the switch. Each application can create its own group for a required LMEP/RMEP combination.

You can associate a group to each LMEP created on a port. There exists a one-to-one relationship between LMEP-port-group. Whenever CFM stops receiving CCMs on this port, it informs a group DOWN event to registered clients like ERPS/EAPS. Whenever CFM starts receiving the CCMs again on this port, a group UP event is sent to registered clients.

Configuring CFM

To configure CFM, create a maintenance domain and assign it a unique MD level. Next, associate MAs with the specified domain and assign MPs within that MA. Optionally, you can configure the transmission interval for the CCMs, destination MAC type for an MA and remote MEPs statically in an MA.

If a MEP fails to receive a CCM before the last advertised TTL value expires, the system logs a message. After the network administrator sees the system log message, he can send a Layer 2 ping and/or a traceroute message to isolate the fault.

Note

CFM does not use *ACL (Access Control List)*; there are no additional ACL entries present for CFM in the `show access-list dynamic` command output.

ExtremeXOS 15.5 provides support for transmitting and receiving ITU-T Y.1731 CCMs. The main difference between IEEE 802.1ag and ITU-T Y.1731 CCMs is between the MAID and MEG ID formats in CCMs:

- MAID ---- MA (format + length + name) with/without MD (format + length +name) details.
- MEG ID ---- MEG (format + length + name) without MD details.
- MA is referred to as MEG in Y.1731 and both are same.
- MA assumes four formats (string/integer/vpn-id/vlan-id) for its name.
- MEG assumes ICC format which is a combination of ICC (6 bytes) + organization specific UMC (6 bytes).
- To support Y.1731 CCMs, an additional name format for MEG name is added for association.

Creating Maintenance Domains

You can create maintenance domains (MDs), or domains, and assign a unique MD level at that time. Available MD levels are numbered from 0 to 7. Higher numerical values are superior MD levels in the CFM hierarchy. Each switch can have a total of eight domains, each with a unique MD level.

You can name domains using any one of the following three formats:

- Simple string—Use an alphanumeric character string with a maximum of 43 characters.
- Domain name server (DNS) name—Use an alphanumeric character string with a maximum of 43 characters.
- MAC address plus 2-octet integer—Use a MAC address and a 2-octet integer. The display format is XX.XX.XX.XX.XX.XX.YYY, where X is the MAC address, and Y is the

2-octet integer. For example, a domain name in this format using 123 as the 16-bit unsigned integer appears as follows: 00:11:22:33:44:55.123.

Note
Whatever convention you choose, you must use that same format throughout the entire domain.

The CFM messages carry the domain name, so the name and naming format must be identical to be understood throughout the domain. You can, however, use different naming conventions on different domains on one switch (up to eight domains allowed on one switch). User-created CFM names are not case sensitive.

- To create a domain and assign an MD level using the DNS convention, use the following command:.

  ```
  create cfm domain dns name md-level level
  ```

- To create a domain and assign an MD level using the MAC address convention, use the following command:.

  ```
  create cfm domain mac mac-addr int md-level level
  ```

- To create a domain and assign an MD level using the string convention, use the following command :

  ```
  create cfm domain string str_name md-level level
  ```

- Although you assign an MD level to the domain when you create that domain, you can change the MD level on an existing domain by running:

  ```
  configure cfm domain domain_name md-level level
  ```

- To delete a domain, use the following command:

  ```
  delete cfm domain domain
  ```

Creating and Associating MAs

Within a given domain, you can associate maintenance associations (MAs). Extreme Networks' implementation of CFM associates MAs with service instances. All of the ports in that service instance are now in that MA and its associated domain.

Each MA associates with one service instance, and each service instance may associate with more than one MA; you can configure more than one MAs in any one domain.

Like the domains, ExtremeXOS supports multiple formats for naming the MA. The following formats are supported for naming the MAs:

- Character string
- 2-octet integer
- RFC 2685 VPN

- VLAN ID
- Y.1731 MEG name

- To add an MA to a domain using the character string format, use the following command:

  `configure cfm domain domain_name add association string name [vlan vlan_name|vman vman_name]`

- To add an MA to a domain using the 2-octet integer format, use the following command:

  `configure cfm domain domain_name add association integer int [vlan vlan_name|vman vman_name]`

- To add an MA to a domain using the RFC 2685 VPN ID format, use the following command:

  `configure cfm domain domain_name add association vpn-id oui oui index index [vlan vlan_name|vman vman_name]`

- To add an MA to a domain using the VLAN ID format, use the following command:

  `configure cfm domain domain_name add association vlan-id vlanid [vlan vlan_name|vman vman_name]`

- To add an MA using Y.1731 compliant MEG, use the following command: `configure cfm domain domain_name add association meg [meg name] [vlan vlan_name|vman vman_name]`

- To delete an MA from a domain, use the following command:

  `configure cfm domain domain_name delete association association_name`

  In addition to supporting multicast destination MAC address for CCM and LTM frames specified by the 802.1ag standard, ExtremeXOS CFM supports the use of a unicast destination address for CCM and LTM frames.

  This allows the support of a CFM operation in a network where use of multicast address is prohibited.

- To configure the destination MAC address type for an MA, use the following command:

  `configure cfm domain domain-name association association_name destination-mac-type [unicast | multicast]`

- Use the following command to add a remote MEP to an MA statically:

  `configure cfm domain domain-name association association_name add remote-mep mepid { mac-address mac_address }`

  ExtremeXOS CFM supports configuring remote MEPs statically for CFM operation where dynamic discovery of MEPs in an MA using multicast address is prohibited.

- To delete a remote MEP from an MA, use the following command:

  `configure cfm domain domain-name association association_name delete remote-mep mepid`

- To configure a remote MEP MAC address, use the following command:

  `configure cfm domain domain-name association association_name remote-mep mepid mac-address mac_address`

Creating MPs and the CCM Transmission Interval

Within an MA, you configure the following MPs:

- Maintenance end points (MEPs), which are one of the following types:
  - UP MEPs—transmit CCMs, and maintain CCM database.
  - DOWN MEPs—transmit CCMs, and maintain CCM database.
- Maintenance intermediate points (MIPs)—pass CCMs through.

Each MEP must have an ID that is unique for that MEP throughout the MA.

- To configure UP and DOWN MEPs and its unique MEP ID, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* `add [` **`end-point`** `[` **`up`** `|` **`down`** `]` **`mepid group`** `{` *`group_name`* `} ] | [` **`intermediate-point`** `]]`

- To change the MEP ID on an existing MEP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* **`end-point`** `[` **`up`** `|` **`down`** `]` **`mepid`** `{` *`mepid`* `}`

- To delete UP and DOWN MEPs, use the following command:

  `configure cfm domain` *`domain_name`* **`association association_name ports`** *`port_list`* `delete [[` **`end-point`** `[` **`up`** `|` **`down`** `]] |` **`intermediate-point`** `]`

- To configure a MIP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* `add [` **`end-point`** `[` **`up`** `|` **`down`** `]` **`mepid group`** `{` *`group_name`* `}] | [` **`intermediate-point`** `]`

- To delete a MIP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* `delete [[` **`end-point`** `[` **`up`** `|` **`down`** `]] |` **`intermediate-point`** `]`

- To configure the transmission interval for the MEP to send CCMs, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* **`end-point`** `[` **`up`** `|` **`down`** `]` **`transmit-interval`** `[` **`3`**`|` **`10`** `|` **`100`** `|` **`1000`** `|` **`10000`** `|` **`60000`** `|` **`600000`** `]`

- To unconfigure the transmission interval for the MEP to send CCMs and return it to the default, use the following command:

  `unconfigure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* **`end-point`** `[` **`up`** `|` **`down`** `]` **`transmit-interval`**

- To enable or disable a MEP, use the following command:

  `configure cfm domain` *`domain_name`* **`association`** *`association_name`* **`ports`** *`port_list`* **`end-point`** `[` **`up`** `|` **`down`** `] [`**`enable`** `|` **`disable`** `]`

Configuring EAPS for CFM Support

- Use the following command to configure *EAPS* for CFM support:

For additional configuration details for CFM support, refer to Configuring CFM on page 1290.

Assigning MEP Group Names to New MEP

To assign MEP Group name when creating a MEP, use the following command:

```
configure cfm domain domain_name association association_name ports
port_list add [[ end-point [ up | down ] mepid group { group_name }] |
[ intermediate-point ]]
```

Assign MEP Group Name to Existing MEP

To assign a MEP Group name to an existing MEP, use the following command:

```
configure cfm domain domain_name association association_name ports
port_list end-point [up|down] [add|delete] group group_name
```

Add a RMEP to MEP Group

To add specific RMEPs for a MEP Group to monitor, use the following command:

```
configure cfm group group_name [add|delete] rmep mepid
```

Monitoring CFM in EAPS

Displaying MEP Groups

To display MEP groups, use the following command:

```
show cfm groups {group_name}
```

```
# sh cfm groups
Group : eapsCfmGrp1     Status : UP
Local MEP       : 11    port  : 41
Remote MEPs     : 10
Client(s)       : eaps
Domain          : MD1
Association     : MD1v2
Group : eapsCfmGrp2     Status : UP
Local MEP       : 12    port  : 31
Remote MEPs     : 13
Client(s)       : eaps
Domain          : MD1
Association     : MD1v2
```

Executing Layer 2 Ping and Traceroute Messages

If the system logs a missed CCM message, the operator can use Layer 2 ping and traceroute messages to isolate the fault. (See Ping and Traceroute for information on how each MP handles these messages.)

Note

You must have all the CFM parameters configured on your network before issuing the ping and traceroute messages.

- To send a Layer 2 ping, use the following command:

  ```
  ping mac mac port port {domain} domain_name {association}
  association_name
  ```

- To send a Link Trace Message (LTM) and receive information on the path, use the following command:

  ```
  traceroute mac mac {up-end-point} port port {domain} domain_name
  {association} association_name {ttl ttl}
  ```

Displaying CFM

To verify your CFM configuration, you can display the current CFM configuration using the `show cfm` command.

The information this command displays includes the total ports configured for CFM, the domain names and MD levels, the MAs and associated service instances, and the UP and DOWN MEPs.

To display the CCM database for each MEP, use the `show cfm detail` command.

Counters are added for missed-hellos, which mainly help to constantly monitor the health of CFM session path and hence tweaking the configured interval times to get the best reliability with the shortest fail over times for radio and wireless networks that rely on CFM to detect any failures. These counters can be displayed using `show cfm session counters missed-hellos` command. The counters can be cleared using "clear counters cfm session missed-hellos" command.

CFM Example

As shown in Figure 159 on page 1295, this example assumes a simple network and assumes that CFM is configured on the access switches, as well as the necessary VMANs configured with the ports added. This example shows a VMAN associated with two maintenance domains and two different MAs. UP MEPs are configured for an MA with MD level 6 and DOWN MEPs are configured for an MA with MD level 3.

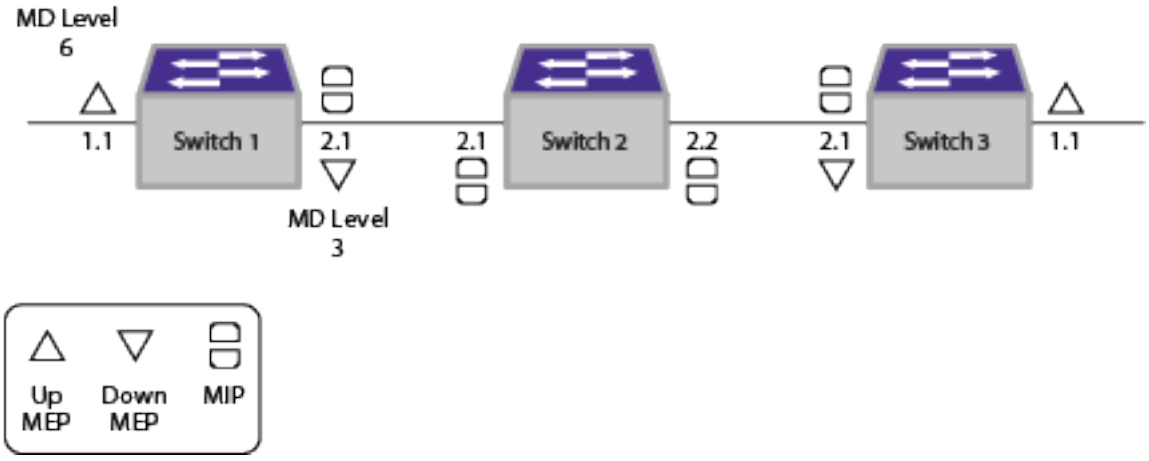


Figure 159: CFM Configuration Example

- Configure switch 1 for this example.

```
create cfm domain string cust-xyz-d6 md-level 6
configure cfm domain cust-xyz-d6 add association string cust-xyz-d6-m100 vman m100
```

```
configure cfm domain cust-xyz-d6 association cust-xyz-d6-m100 port 1:1 add end-point
up 10
configure cfm domain cust-xyz-d6 association cust-xyz-d6-m100 port 2:1 add
intermediate-point
create cfm domain string core-d3 md-level 3
configure cfm domain core-d3 add association string core-d3-m100 vman m100
configure cfm domain core-d3 association core-d3-m100 port 2:1 add end-point down 10
```

- Configure switch 2 for this example.

```
create cfm domain string core-d3 md-level 3
configure cfm domain core-d3 add association string core-d3-m100 vman m100
configure cfm domain core-d3 association core-d3-m100 port 2:1 add intermediate-point
configure cfm domain core-d3 association core-d3-m100 port 2:2 add intermediate-point
```

- Configure switch 3 for this example.

```
create cfm domain string cust-xyz-d6 md-level 6
configure cfm domain cust-xyz-d6 add association string cust-xyz-d6-m100 vman m100
configure cfm domain cust-xyz-d6 association cust-xyz-d6-m100 port 1:1 add end-point
up 20
configure cfm domain cust-xyz-d6 association cust-xyz-d6-m100 port 2:1 add
intermediate-point
create cfm domain string core-d3 md-level 3
configure cfm domain core-d3 add association string core-d3-m100 vman m100
configure cfm domain core-d3 association core-d3-m100 port 2:1 add end-point down 20
```

- To display the group database, use the following command:

```
show cfm groups
```

Y.1731—Compliant Performance Monitoring

Compliant performance monitoring is based on the ITU-T Y.1731 standard and deals with the Ethernet Delay Measurement (ETH-DM) function and Ethernet Frame-Loss Measurement (ETH-LM).

Frame-Delay Measurement

ExtremeXOS software supports:

- Two-way delay measurement—Delay Measurement Message (DMM) and Delay Measurement Reply (DMR).
- Continuous (proactive) measurement of frame delay and frame delay variation.
- On-demand measurement of frame delay and frame delay variation.

By default, DMM is not enabled. You must explicitly enable the DMM transmission for a CFM segment, either as continuous or on-demand mode.

A network interface is considered attached to a subnetwork. The term *segment* is used to refer to such a subnetwork, whether it be an Ethernet LAN, a ring, a WAN link, or even an SDH virtual circuit.

Frame-Delay measurement is done between two specific end points within an administrative domain. Frame delay and frame delay variation measurements are performed in a maintenance association end point (MEP) by sending and receiving periodic frames with ETH-DM information to and from the peer end point during the diagnostic interval.

When a CFM segment is enabled to generate frames with ETH-DM information, it periodically sends frames with ETH-DM information to its peer in the same maintenance association (MA) and expects to receive frames with ETH-DM information from its peer in the same MA.

The following list offers specific configuration information that is required by a peer to support ETH-DM:

- Maintenance domain (MD) level—The MD level at which the peer exists.
- Priority—The priority of the frames with ETH-DM information.
- Drop eligibility—Frames with ETH-DM information that are always marked as drop ineligible.
- Transmission rate.
- Total transmit interval.

A node transmits frames with ETH-DM information with the following information element:

- TxTimeStampf: Timestamp at the transmission time of the ETH-DM frame.
- RxTimeStampb: Timestamp at which the switch receives the DMR back.

  Whenever a valid DMM frame is received by the peer, a DMR frame is generated and transmitted to the requesting node.

- A DMM frame with a valid MD level and a destination MAC address equal to the receiving node's MAC address is considered to be a valid DMM frame. Every field in the DMM frame is copied to the DMR frame with the following exceptions:
  - The source and destination MAC addressed are swapped.
  - The OpCode field is changed from DMM to DMR.

The switch makes two-way frame delay variation measurements based on its ability to calculate the difference between two subsequent two-way frame delay measurements.

To allow a more precise two-way frame delay measurement, the peer replying to the frame with ETH-DM request information may include two additional timestamps in the ETH-DM reply information:

- RxTimeStampf—Timestamp at the time of receiving a frame with ETH-DM request information
- TxTimeStampb—Timestamp at the time of transmitting a frame with ETH-DM reply information

Here the frame delay is calculated by the peer that receives the DMR as follows:

- Frame Delay = (RxTimeStampb - TxTimeStampf) - (TxTimeStampb - RxTimeStampf)

Figure 160 describes the DMM and DMR message flows between two end points.

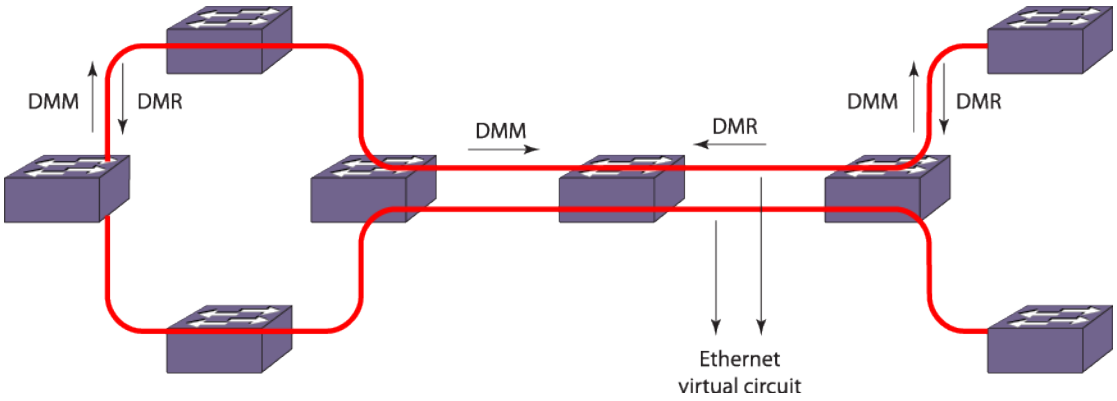


Figure 160: Two-Way Frame Delay and Frame Delay Variance Measurement

The PDUs used to measure frame delay and frame delay variation are the DMM and the DMR PDUs where DMM is initiated from a node as a request to its peer and DMR is the reply from the peer.

If you try to enable the transmission for a CFM segment whose configuration is not complete, the trigger is rejected and an error message similar to the following is given:

```
ERROR: CFM Configuration is not complete for segment "s1" to start transmission
```

Note
A CFM segment without a domain and an association is considered to be an incomplete segment.

Upon enabling the transmission from a CFM segment, the segment transmits DMM frames, one at each transmit-interval which is configured through the CLI. If the user enables on-demand transmission, the segment transmits "X" number of DMMs and moves back to the disabled state, where "X" is the number of frames specified by the user through the CLI.

For continuous transmission, the segment continues to transmit DMM frames until stopped by the user. This transmission continues even after reboot for both continuous and on-demand mode. For on-demand transmission, the segment, which was enabled to transmit "X" number of frames, and is still transmitting, starts transmitting again "X" number of frames after reboot, or failover, or process restart. The old statistics are not preserved for both continuous and on-demand mode for all the above three scenarios.

Upon transmitting a DMM, the segment is expected to get a reply from the destination within the specified time. If a reply is received after that time, that reply will be considered as a delayed one.

If a reply is not received within the transmit-interval, that is, between two subsequent DMM transmissions, then that frame is considered as lost. Once the percentage of the sum of lost and delayed frames reaches the alarm threshold, an alarm is generated and the segment is moved to the alarming state. This state is maintained until the percentage of valid replies reaches the clear threshold. These alarm and clear states are maintained for a specified window, which holds a set of recent frames and their corresponding delays.

Various times are recorded at the segment level during the transmission of DMM frames.

- Start time—Time at which the segment started the current transmission.
- Min delay time—Time at which the minimum delay occurred in the current transmission window.
- Max delay time—Time at which the maximum delay occurred in the current transmission window.
- Alarm time—The recent alarm time, if any, during the current transmission.

The mean delay and delay variance for the current window is also measured whenever the user polls the segment statistics.

Frame-Loss Measurement

Frame-loss is measured by sending and receiving frames with frame-loss information between peer maintenance end points (MEPs).

Frame-loss ratio is defined as a percentage of the number of service frames not delivered divided by the total number of service frames during a defined time interval, where the number of service frames not delivered is the difference between the number of service frames arriving at the ingress Ethernet flow point and the number of service frames delivered at the egress Ethernet flow point in a point-to-point Ethernet connection (see Figure 161).

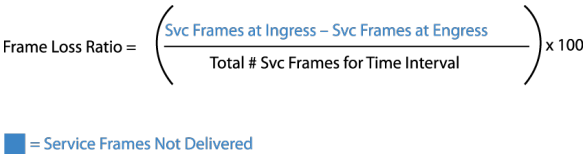


Figure 161: Frame-Loss Ratio Formula

To support frame-loss measurement, a MEP requires the following configuration information:

- Maintenance domain (MD) level—MD level at which the MEP exists.
- Frame-loss measurement transmission period—time interval when frame-loss measurement frames are sent.
- Priority—identifies the priority of the frames with frame-loss measurement information (configurable per operation).
- Drop eligibility—frames with frame-loss measurement information are always marked as drop ineligible (not necessarily configured).

A maintenance intermediate point (MIP) is transparent to frames with frame-loss measurement information. Therefore MIPs do not require any information to support frame-loss measurement functionality.

There are two frame-loss measurement methods:

- Dual-Ended Frame-Loss Measurement
- Single-Ended Frame-Loss Measurements

Dual-Ended Frame-Loss Measurement

Dual-ended frame-loss measurement is a form of proactive OAM for performance monitoring and is useful for fault management.

Note

ExtremeXOS does not support dual-ended frame-loss measurement.

MEPs send periodic dual-ended frames with frame-loss measurement information to peer MEPs in a point-to-point MD. Each MEP terminates the dual-ended frames with frame-loss measurement information and makes the near-end and far-end loss measurements. Near-end frame loss refers to frame loss associated with ingress data frames, while far-end frame loss refers to frame loss associated with egress data frames. This function is used for performance monitoring at the same priority level as used for CCM.

The protocol data unit (PDU) for dual-ended frame-loss measurement information is Continuity Check Message (CCM).

Single-Ended Frame-Loss Measurements

Single-ended frame-loss measurement facilitates on-demand OAM. MEPs carry out frame-loss measurments by sending frames to peer MEPs with frame-loss measurement request information and receiving frames with frame-loss measurement reply information. This Y_1731 frame-loss measurement depends on VLAN statistics.

The PDU for single-ended frame-loss measurement requests is Loss Measurement Message (LMM). The PDU for single-ended frame-loss measurement reply is Loss Measurement Reply (LMR). Figure 162 shows the transmission of LMM and LMR for frame-loss measurement.

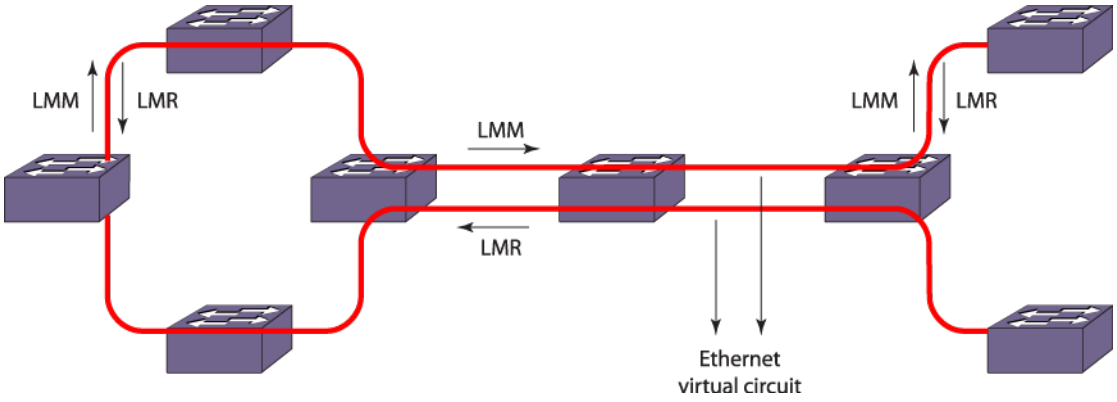


Figure 162: Two-Way Frame-Loss Measurement

A MEP maintains two local counters for each peer MEP it is monitoring for frame-loss:

- TxFCl—in-profile data frames transmitted to the peer MEP.
- RxFCl—in-profile data frames received from the peer MEP.

For an on-demand loss measurement, a MEP periodically transmits LMM frames with TxFCf (value of the local TxFCl counter at the time of LMM frame transmission). Upon receiving a valid LMM frame, a MEP sends an LMR frame to the requesting MEP.

(Valid LMM frames have a valid MD level and a destination MAC address equal to the receiving MEP's MAC address.)

An LMR frame contains the following values:

- TxFCf—TxFCf value copied from the LMM frame.
- RxFCf—RxFCl value when the LMM frame was received.
- TxFCb—TxFCl value when the LMR frame was transmitted.

Upon receiving an LMR frame, a MEP uses the following values to make near-end and far-end loss measurements:

- Received LMR frame's TxFCf, RxFCf, and TxFCb values, and local counter RxFCl value at the time this LMR frame was received. These values are represented as TxFCf[tc], RxFCf[tc], TxFCb[tc], and RxFCl[tc]; where tc is the time the current reply frame was received.
- Previous LMR frame's TxFCf, RxFCf, and TxFCb values, and local counter RxFCl value at the time the previous LMR frame was received. These values are represented as TxFCf[tp], RxFCf[tp], TxFCb[tp], and RxFCl[tp],where tp is the time the previous reply frame was received.

Far-End Frame Loss = (TxFCf[tc] – TxFCf[tp]) – (RxFCf[tc] – RxFCf[tp])

Near-End Frame Loss = (TxFCb[tc] – TxFCb[tp]) – (RxFCl[tc] – RxFCl[tp])

Availability Time and Severly Errored Seconds (SES)

Frame loss is measured by sending and receiving frames with frame-loss information between peer MEPs.

Each MEP performs frame-loss measurements which contribute to unavailable time. Since a bidirectional service is defined as unavailable if either of the two directions is declared unavailable, frame-loss measurement must facilitate each MEP to perform near-end and far-end frame loss measurements.

Near-end frame loss refers to frame loss associated with ingress data frames, while far-end frame loss refers to frame loss associated with egress data frames. Both near-end and far-end frame loss measurements contribute to near-end severely errored seconds (near-end SES) and far-end severely errored seconds (far-end SES) respectively, which together contribute to unavailable time.

A period of unavailable time begins at the onset of x consecutive Severely Errored Seconds (SES) events. These x seconds are part of unavailable time. A new period of available time begins at the onset of x consecutive non-SES events. These x seconds are part of available time.

A SES is declared when, during one measurement period, the number of frames lost exceeds a threshold. ExtremeXOS logs the start and end time of the unavailable periods (see Figure 163 from ITU-T G.7710).

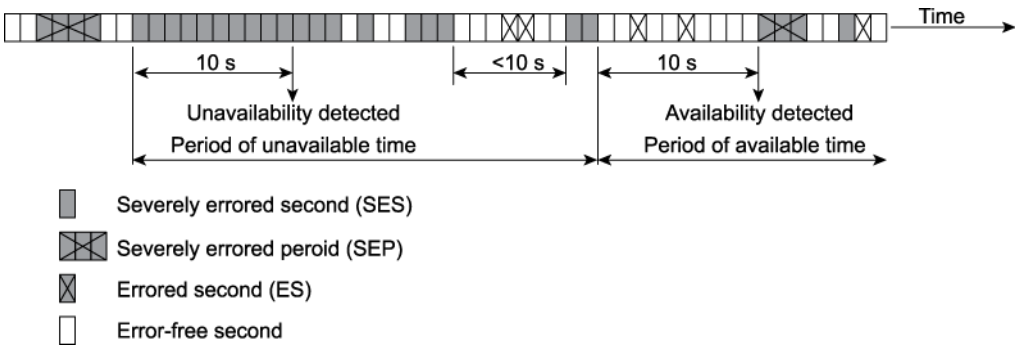


Figure 163: SES

Configuring a CFM Segment

Use the following commands to configure a CFM segment.

Some of these commands are optional and, if not configured, the default values are used. Table 84 lists the default values for delay measurement for a CFM segment.

Table 84: Default Values for Delay Measurement for a CFM Segment

Configuration	Default Values
Transmit interval	10 seconds
Window	60 frames
Timeout	50 milliseconds
Alarm threshold	10%
Clear threshold	95%
Dot1p priority	6

Table 85 lists the default values for loss measurement for a CFM segment.

Table 85: Default Values for Loss Measurement for a CFM Segment

Configuration	Default Values
LMM Transmit interval	90 seconds
Dot1p priority	6
Window	1200 frames
SES threshold	0.01
Consecutive available count	4

Note
The statistics for a particular transmission are preserved until the user triggers the transmission once again or if `clear counters cfm segment` is triggered from the CLI.

Managing a CFM Segment

You can create, delete, add CFM segments.

- To create a CFM segment, use the following command:

  ```
  create cfm segment segment_name destination mac_addr {copy
  segment_name_to_copy}
  ```

- To delete a CFM segment, use the following command:

  ```
  delete cfm segment [segment_name | all]
  ```

- To add a CFM domain to a CFM segment, use the following command:

  ```
  configure cfm segment segment_name add domain domain_name association
  association_name
  ```

- To delete a CFM domain from a CFM segment, use the following command:

  ```
  configure cfm segment segment_name delete domain association
  ```

- To configure the transmission interval between two consecutive DMM or two consecutive LMM frames, use the following command:

  ```
  configure cfm segment segment_name {frame-delay | frame-loss}
  transmit-interval interval
  ```

  The same transmit-interval is used for both delay and loss measurements.

- To get separate values for delay and loss measurements, use the following command:

  ```
  configure cfm segment frame-delay/frame-loss transmit interval
  interval
  ```

- To configure the dot1p priority of a DMM frame, use the following command:

  ```
  configure cfm segment segment_name frame-delay dot1p dot1p_priority
  ```

- To configure the dot1p priority of a LMM frame, use the following command:

  ```
  configure cfm segment segment_name frame-loss dot1p dot1p_priority
  ```

- To configure the dot1p priority of the CFM segment, use the following command:

  ```
  configure cfm segment segment_name dot1p dot1p_priority
  ```

  The same priority is used for both delay and loss measurements.

- To get separate values of priority for delay and loss measurements, use the following command:

  ```
  configure cfm segment segment_name frame-delay dot1p dot1p_priority

  configure cfm segment segment_name frame-loss dot1p dot1p_priority
  ```

- To configure the alarm and clear threshold value for CFM segment, use the following command:

  ```
  configure cfm segment segment_name [alarm-threshold | clear-threshold]
  value
  ```

- To configure the window size to be used for calculating the threshold values, use the following command:

  ```
  configure cfm segment segment_name window size
  ```

  The same window size is used for both delay and loss measurements.

- To get separate values of window size for delay and loss measurements, use the following:

  ```
  configure cfm segment segment_name frame-loss window window_size

  configure cfm segment segment_name frame-delay window window_size
  ```

- To configure the window size of a DMM frame to be used for calculating the threshold values, use the following command:

  ```
  configure cfm segment segment_name frame-delay window window_size
  ```

- To configure the window size of a LMM frame to be used for calculating the threshold values, use the following command:

  ```
  configure cfm segment segment_name frame-loss window window_size
  ```

- To trigger DMM frames at the specified transmit interval, use the following command:

  ```
  enable cfm segment frame-delay measurement segment_name mep mep_id
  [continuous | count ] value
  ```

- To disable the transmission of the DMM frames for a particular CFM segment, use the following command:

  ```
  disable cfm segment frame-delay measurement segment_name mep mep_id
  ```

- To show the configuration and status of a specific CFM segment, use the following command:

  ```
  show cfm segment {segment_name}
  ```

- To show the configuration and status of a specific CFM segment doing delay measurement, use the following command:

  ```
  show cfm segment frame-delay {segment_name]}
  ```

- To show the configuration and status of a specific CFM segment doing loss measurement, use the following command:

  ```
  show cfm segment frame-loss {segment_name]}
  ```

- To display the frame delay statistics for the CFM segment, use the following command:

  ```
  show cfm segment frame-delay statistics {segment-name}
  ```

- To configure the timeout value for a CFM segment, use the following command:

  ```
  configure cfm segment segment_name timeout msec
  ```

- To add or delete the local MEP for a given CFM segment, use the following command:

  ```
  configure cfm segment segment_name frame-loss [add|delete] mep mep_id
  ```

- To set the percentage of frames lost in a measurement period so that it will be marked as SES (severely errored second), use the following command:

  ```
  configure cfm segment segment_name frame-loss ses-threshold percent
  ```

- To set the number of consecutive measurements used to determine the availability status of a CFM segment, use the following command:

  ```
  configure cfm segment segment_name frame-loss consecutive frames
  ```

- To start the transmission of LMM frames for the set transmit interval, use the following command:

  ```
  enable cfm segment frame-loss measurement segment_name mep mep_id
  [continuous | count frames]
  ```

  Note
  For the above command, If the the segment is not completely configured, frames are not transmitted and an error occurs.

- To stop the transmission of the LMM frames for a particular CFM segment, use the following command:

  ```
  disable cfm segment frame-loss measurement segment_name mep mep_id
  ```

- To display the frame loss or frame delay statistics for the CFM segment, use the following command:

  ```
  show cfm segment {{segment_name} | {frame-delay {segment_name}} |
  {frame-loss {segment_name {mep mep_id}}}}
  ```

  Note
  Frame-loss measurements are not supported on platforms where the *VLAN* packet statistics are not retrieved, and on up-meps.

Clearing CFM Information

- To clear cfm segment counters, use the following commands:

  ```
  clear counters cfm segment segment_name
  clear counters cfm segment all
  ```

- To clear cfm segment counters specific to DMM, use the following command:

  ```
  clear counters cfm segment segment_name frame-delay
  ```

- To clear cfm segment counters specific to LMM, use the following commands:

  ```
  clear counters cfm segment segment_name frame-loss
  clear counters cfm segment segment_name frame-loss mep mep_id
  ```

Y.1731 MIB Support

ExtremeXOS 15.5 supports Y.1731 performance measurement MIB defined by MEF - 36. The performance monitoring process is made up of a number of performance monitoring instances, known as performance monitoring (PM) sessions. A PM session can be initiated between two MEPs in a MEG and be defined as either a loss measurement (LM) session or delay measurement (DM) session. The LM session can be used to determine the performance metrics frame loss ratio, availability, and resiliency. The DM session can be used to determine the performance metrics Frame Delay.

The MIB is divided into a number of different object groupings: the PM MIB MEP objects, PM MIB loss measurement objects, PM MIB delay measurement objects, and SOAM PM notifications. The initial implementation of MIB supports GET operations for Frame Loss& Frame Delay.

MIB Specific Data

- A measurement interval of 15 min to be supported
- Default message period/transmit-interval of LMMs is 1 sec (Min = 1sec in current CLI) * Default message period/transmit-interval of DMMs is 100msec (Min = 1 sec in current CLI)
- Repetition Time can be set to 0 which means that there is no gap between measurement intervals
- Number of History measurement intervals can be 2-1000, though it expects at least 32 measurement intervals to be stored and 96 are recommended.
- Both DM and LM sessions are MEP to MEP sessions. The index of all the DM/LM tables includes MD, MA, MEP table indices as well as the particular DM/LM session.
- Currently DM sessions are not MEP-to MEP based but only segment based sessions. To support DM tables in the MIB, changes are required in the current CLI & backend delay implementation.

Limitations

- Currently we are storing a maximum of 1800 frames data for each LMM/DMM session. But to support at least 2 history and 1 current measurement intervals, we need to store 2700 frames (if message period is 1 sec, Repetition time is 0, measurement interval is 15 min) for each delay/loss session.
- Each frame's data is about 60 bytes for LMM and which takes about 44 MB of memory for 288 sessions

EFM OAM—Unidirectional Link Fault Management

Unidirectional Link Fault Management

With EFM OAM, certain physical layers can support a limited unidirectional capability.

The ability to operate a link in a unidirectional mode for diagnostic purposes supports the maintenance objective of failure detection and notification. Unidirectional OAM operation is not supported on some legacy links but is supported on newer links such as 100BASE-X PCS, 1000BASE-X PCS, and 10GbE RS. On technologies that support the feature, OAM PDUs can be transmitted across unidirectional links to indicate fault information. To the higher layers, the link is still failed in both directions, but to the OAM layer, some communication capabilities exist. The distinction between a unidirectional link and a normal link is shown in Figure 164.

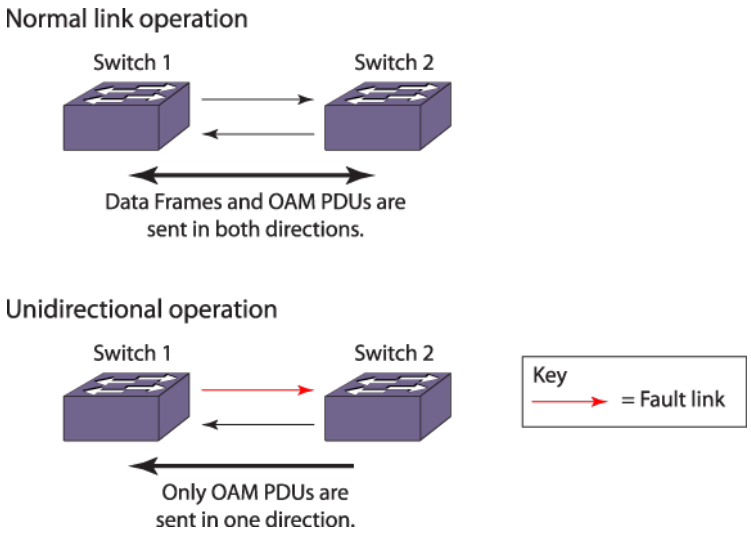


Figure 164: Normal Link and Unidirectional Operation

You can enable unidirectional link fault detection and notification on individual ports with CLI commands. This allows appropriate register settings to transmit OAM PDUs even on a link that has a slowly deteriorating quality receive path or no receive path at all. Then, when a link is not receiving a signal from its peer at the physical layer (for example, if the peer's laser is malfunctioning), the local entity can set a flag in an OAM PDU to let the peer know that its transmit path is inoperable.

The operation of OAM on an Ethernet interface does not adversely affect data traffic because OAM is a slow protocol with very limited bandwidth potential, and it is not required for normal link operation. By utilizing the slow protocol MAC address, OAM frames are intercepted by the MAC sub layer and cannot propagate across multiple hops in an Ethernet network. This implementation assures that OAM PDUs affect only the operation of the OAM protocol itself and not user data traffic.

The IEEE 802.3ah standard defines fault notifications based on one-second timers. But by sending triggered OAM PDUs on detecting link down/local fault rather that waiting to send on periodic PDUs, failure detection is less than one second can be achieved, thereby accelerating fault recovery and network restoration.

EFM OAM uses standard length Ethernet frames within the normal frame length of 64 to 1518 bytes as PDUs for their operation. Table 86 describes the fields of OAM PDUs.

Table 86: OAM PDU Fields

Field	Octets	Description	Value
Destination Address	6	Slow protocol multicast address	01:80:C2:00:00:02
Source Address	6	Port's individual MAC address	Switch MAC
Length/Type	2	Slow protocol type	0x8809
Subtype	1	Identifies specific slow protocol	0x03

Table 86: OAM PDU Fields (continued)

Field	Octets	Description	Value
Flags	2	Contains status bits	see the following figure
Code	1	Identifies OAM PDU type	0x00 (Information TLV)
Data/Pad	42-1496	OAM PDU data	0x00 (END of TLV)
FCS	4	Frame check sequence	

Configuring Unidirectional Link Fault Management

To configure unidirectional link fault management on a port or ports, use the following command:

```
enable ethernet oam ports [port_list | all] link-fault-management
```

To clear the counters on a configured port, use the following command:

```
clear ethernet oam {ports [port_list} counters
```

To unconfigure unidirectional link fault management, use the following command:

```
disable ethernet oam ports [port_list | all] link-fault-management
```

To display the Ethernet OAM settings, use the following command:

```
show ethernet oam {ports [port_list} {detail}
```

When configured, the following behavior on the port is observed:

- A log indicates that traffic on the port is blocked.
- All received traffic on that port is blocked except for Ethernet OAM PDUs.
- To higher layers, a failure is reported as a link down but OAM can use the link to send OAM traffic.

Two-Way Active Measurement Protocol

The Two-Way Active Measurement Protocol, defined in RFC 5357, specifies an industry standard for measuring round-trip performance between two devices that support the TWAMP protocols. TWAMP defines two protocols; the TWAMP-Control protocol and the TWAMP-Test protocol. The TWAMP-Control protocol is used to setup test sessions. The test sessions use the TWAMP-Test protocol to transmit and reflect performance measurement packets. The TWAMP-Control protocol utilizes TCP for communication, while the TWAMP-Test protocol utilizes UDP.

Note
Only the TWAMP-Test protocol is supported.

The test sessions are setup via the ‘Request-Session’ command message, sent from the Control-Client to the Server. The Server replies with an ‘Accept Session’ message, which indicates if the Server is capable of accommodating the request or not. The Control-Client may send several ‘Request-Session’ command messages to setup multiple test sessions. To begin the tests, the Control-Client transmits a ‘Start-Session’ command message. The Server replies with a ‘Start-Ack’ message. The Control-Client does not begin its test until it receives the ‘Start-Ack’ message. This allows the Server ample time to configure the test sessions. The Control-Client will stop the test sessions with a ‘Stop-Sessions’ message. The Server does not respond to this message.

The ExtremeXOS 16.1 implementation of TWAMP consists of the development of the TWAMP logical role of the Session-Reflector. The user is required to configure endpoints, which define the destination of TWAMP-Test packets generated by the client. An endpoint receiving a new TWAMP-test packet creates a test session consisting of the following four-part tuple; client IP address, client UDP port, endpoint IP address, and endpoint UDP port. The tuple does not include the VR because it requires the default VR for the first phase. A session timeout value, configured globally, determines the amount of time test sessions exist after the last reception of a TWAMP-Test packet. Test sessions are used to keep track of the session data, such as the sequence number. The Session-Reflector will still respond to TWAMP-Test packets that do not match an existing test session or if a new test session cannot be created due to lack of resources.

TWAMP-Test Protocol

The TWAMP logical roles of the Session-Sender and Session-Reflector use the TWAMP-Test protocol. The Session-Sender, controlled by the Client, transmits TWAMP-Test packets to the Session-Reflector. The Session-Reflector processes the packet, copies the fields from that packet into the reply, and then transmits a reply packet back to the Session-Sender. The reply packet is larger than the request packet unless the Clients adds padding.

The implementation of the Session-Reflector relies on the user to create endpoints, which open up available ports for reflecting TWAMP-Test packets. The endpoint configuration acts as a control list to limit the number of locally opened sockets. The user may choose to use authentication or encryption to protect the transmitted data from nefarious entities. The shared keys used for the TWAMP-Test derive from the TWAMP 3-way handshake that establishes the control session. Both devices must configure matching shared keys.

Bidirectional Forwarding Detection (BFD)

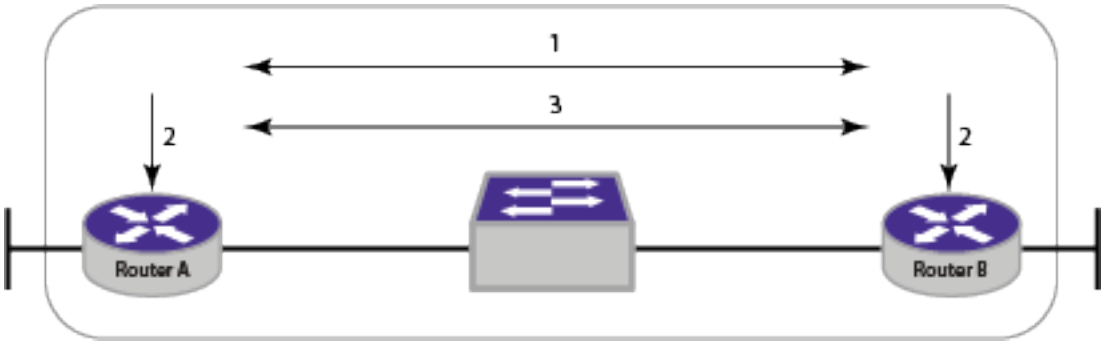
BFD Overview

Bidirectional Forwarding Detection (BFD) is a hello protocol that provides the rapid detection of failures in the path and informs the clients (routing protocols) to initiate the route convergence.

It is independent of media, routing protocols, and data protocols. BFD helps in the separation of forwarding plane connectivity and control plane connectivity.

Different routing protocol hello mechanisms operate in variable rates of detection, but BFD detects the forwarding path failures at a uniform rate, thus allowing for easier network profiling and planning, and consistent and predictable re-convergence time.

Figure 165 shows a BFD topology.



1. Routing protocol (BFD client) discovers neighbor.
2. Routing protocol informs BFD to create session with neighbor.
3. BFD establishes session with neighbor.

Figure 165: BFD Topology

The routing protocols first learn the neighbor and make entries in the forwarding table. Then protocols can register the neighbor address with BFD and ask to monitor the status of the path. BFD establishes the session with a remote BFD and monitors the path status.

You can configure detection multipliers and TX and RX intervals on a directly connected interface (*VLAN*).

- The detection multiplier signifies the number of BFD packets the BFD server waits for after which a timeout is declared.
- The receive interval is the interval at which the BFD server is ready to receive packets.
- The transmit interval is the interval at which the BFD server is ready to transmit packets.

For example, when two nodes, A and B, initiate a BFD session between them, a negotiation about the receive and transmit intervals occurs.

The receive interval of node A is calculated as the maximum of the configured receive interval of node A and the configured transmit interval of node B. The same applies to node B.

If multiple clients ask for the same neighbor on the same interface, then a single BFD session is established between the peers.

Figure 166 shows the behavior when a failure occurs.

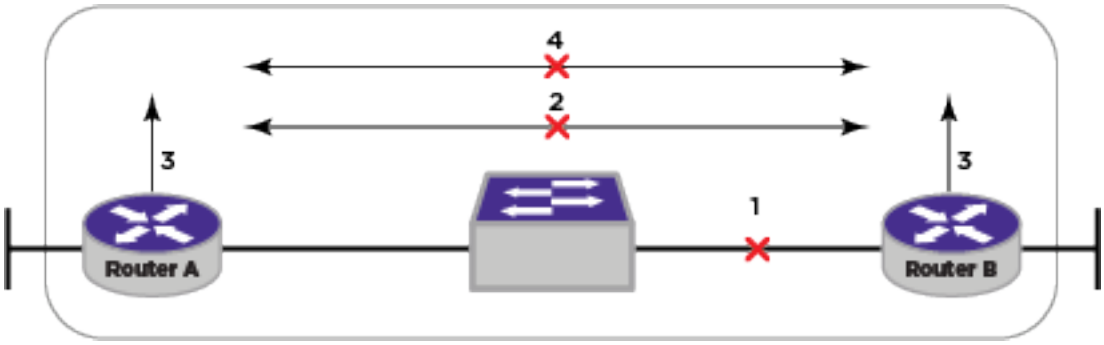


Figure 166: BFD Failure Detection

BFD detects the failure first, and then informs the registered clients about the neighbors.

BFD operates in an asynchronous mode in which systems periodically send BFD control packets to one another. If a number of those packets in a row are not received by the other system, the session is declared to be down.

Simple password authentication can be included in the control packet to avoid spoofing.

BFD for OTM, BGP, and hitless failover are supported.

This feature is available on all platforms.

Note
BFD can be used to protect IPv4 & IPv6 static routes, OSPFv2 & *OSPFv3 (Open Shortest Path First version 3)* interfaces and *BGP (Border Gateway Protocol)* and *MPLS (Multiprotocol Label Switching)* interfaces. For more information, see Configuring Static Routes on page 908, BFD for OSPF on page 1392, or refer to Managing the MPLS BFD Client on page 1116.

Limitations

The following limitations apply to BFD in this release:

- BGP, OSPF (Open Shortest Path First), MPLS, OTM, and static routes act as BFD clients.
- The echo function is not supported.
- Hardware assist BFD is supported only on standalone X460-G2, X465, X590, and X695 series switches.
- Hardware assist BFD is not supported Universal platforms.
- The number of sessions handled by BFD at minimal timers (less than 100ms) varies depending on the platform type, the implementation type (H/w or S/w), and the processing load (which is influenced by other protocols being enabled or other system conditions such as software forwarding).
- The following scaling limits apply to BFD hardware assist:
  - Maximum 900 sessions if PTP feature not enabled.

- Maximum 425 sessions with PTP enabled.
- Maximum of 256 sessions with 3 ms transmit interval.

Configuring BFD

You can enable, disable, configure, and unconfigure BFD.

- To enable or disable BFD, use the following command:

  ```
  [enable | disable] bfd vlan vlan_name
  ```

- To configure the detection multipliers and TX and RX intervals, use the following command:

  ```
  configure bfd vlan vlan_name [{detection-multiplier multiplier}
  {receive-interval rx_interval} {transmit-interval tx_interval}]
  ```

- To specify either authentication using a simple password or no authentication, use the following command:

  ```
  configure bfd vlan vlan_name authentication [none | simple-password
  {encrypted} password]]
  ```

- To unconfigure BFD, use the following command:

  ```
  unconfigure bfd vlan vlan_name
  ```

Configuring BFD Intervals

In case of BFD implemented in software, the ability to run low-range sub-second intervals depends on the CPU load, which in turn depends on other protocol configurations and system conditions, such as software forwarding. It also depends on switch model.

To find a suitable low-range interval for the chosen configuration and network load, statistics of missed hello packets can help. These hello packets are exchanged between BFD peers, and if *n* number of packets are missed in a row, path failure will be declared, where *n* is detection-multiplier. However, missed hellos less than *n* number will not declare failure, but if statistics are steadily increasing, it can indicate that the currently chosen BFD interval is not appropriate.

In that case, a next longer interval/ higher detection multiplier can be chosen for the BFD session, and statistics can be monitored again. In addition, when statistics are observed periodically, missed hellos can give an indication of whether health of the path is deteriorating. These statistics are collected continuously for each BFD session. The commands to display the statistics of missed hellos are provided in Displaying BFD Information on page 1313.

Setting Strict BFD Session Protection

If the BFD session is down, but BFD protected static route is still in the routing table after reboot, the BFD session is never established, because during reboot, the BFD session is in the INIT state, and the static route is brought up without considering BFD session state. This situation can cause traffic loss because the link to the gateway actually is down. This command turns down the static route during reboot if BFD session is in the INIT state. This behavior is different from other BFD clients (such as *OSPF (Open Shortest Path First)*) in the same INIT situation.

To enable strict BFD session protection, use the following command:

```
enable iproute {protection} bfd strict
```

Note
Enabling strict BFD generates a warning message to reboot the switch so as to update the route table: `WARNING: Please reboot the switch for the strict BFD to take effect.`

To disable strict BFD session protection, use the following command:

```
disable iproute {protection} bfd strict
```

To show strict BFD session protection status, use the following command:

```
show iproute bfd
```

Displaying BFD Information

The following commands display information regarding BFD configuration and process.

- To display information on BFD sessions, use the following command:
  `show bfd`
- To display information on BFD global counters, use the following command:
  `show bfd counters`
- To display information on BFD session counters, use the following command:
  `show bfd session {ipv4 | ipv6} {ipaddress} counters {vr [vrname | all]}`
- To display the configuration of a specific interface or those specific counters, use the following command:
  `show bfd vlan {vlan_name}`
- To display the counters of a specific interface, use the following command:
  `show bfd vlan {vlan_name} counters`
- To display the session status of a particular client, use the following command:
  `show bfd session client [ bgp {ipv4 | ipv6} | mpls | | static {ipv4 | ipv6}] {vr [vrname | all]}`
- To display the session status information for all VRs, use the following command:
  `show bfd session {ipv4 | ipv6} {ipaddress } { vr [vrname |all ] }`
- To display session status information in detail for all VRs, use the following command:
  `show bfd session {ipv4 | ipv6} {ipaddress } detail {vr [vrname | all]}`
- To display counters of missed hello packets for all available sessions, use the following command:
  `show bfd session counters missed-hellos {session-id first {- last} | neighbor ipaddress {vr [vrname | all]} | vr [vrname| all]} {detail | no-refresh | refresh}`

- To display counters of missed hello packets for a session using neighbor IP address, use the following command:

  `show bfd session counters missed-hellos neighbor ipaddress {vr [vrname | all]}`
- To display counters of missed hello packets for a particular range of session ID, use the following command:

  `show bfd session counters missed-hellos session-id first {- last}`
- To show strict BFD session protection status, use the following command:

  `show iproute bfd`

Clearing BFD Information

To clear global, session, or interface counters, use the following command:

`clear counters bfd {session | interface | missed-hellos}`

To clear missed hellos counters for a session using neighbor IP address:

`clear counters bfd missed-hellos neighbor ipaddress {vr [vrname | all]}`

To clear missed hellos counters for a particular range of session ID:

`clear counters bfd missed-hellos {session-id first {- last}`

BFD Hardware Assist

Bidirectional Forwarding Detection (BFD) hardware assist support provides the functionality to run a BFD session in hardware. Effective failure detection requires BFD to run at high frequencies (using aggressive timers as low as 3 ms), which was not possible in the Software Mode because of CPU and ExtremeXOS restrictions.

Starting with ExtremeXOS 30.5, BFD hardware assist supports multi-hop.

To make BFD sessions run in the hardware, the following configuration is required:

- Hardware BFD is enabled by configuring one of the unused front panel port as a loopback port, which should not be available for switching user data traffic. This port is used internally by the BFD hardware to send control packets.
- Ensure IP forwarding is enabled on the BFD interfaces.
- The next hop MAC address of the neighbor should be known for the session creation. The BFD process triggers ARP to resolve the next hop MAC address, if this is not configured statically.

L2 topology convergence time might affect the BFD sessions running at the shortest interval.

Unconfiguring loopback for BFD disables the hardware assist support. The supported shortest interval is 3 ms in hardware mode.

Sample configuration:

Configure the port 1 as loopback port dedicated for BFD:

```
#configure bfd hardware-assist primary loopback-port 1
```

To check the number of BFD sessions supported in the hardware and configured loopback port:

```
#show bfd
..
..
Hardware Assist Operational State         : Enabled(Loopback
port not configured)
Hardware Assist Primary Loopback Port     : 1
Hardware Assist Secondary Loopback Port   : None
Maximum # of Hardware Assist Sessions     : 900
```

To find the session running the hardware:

```
#show bfd session detail
Neighbor       : 10.10.10.2          Local       : 10.10.10.1
VR-Name        : VR-Default          Interface   : v1
Session Type   : Single Hop          State       : Down
..
Hardware Assist                 : Yes
```

To see the hardware counters:

```
#show bfd counters
#show bfd session 10.10.10.2 counters
```

BFD MIB Table Support

ExtremeXOS 15.5 supports read-only for all BFD MIB tables, global objects, and supports BFD notifications as well. BFD-MIB implementation is based on draft-ietf-bfd-mib-14, and draft-ietf-bfd-tc-mib-02. Currently, the BFD MIB is kept under the enterprise MIB in ExtremeXOS implementation.

The SET operation is supported only for MIB object 'bfdSessNotificationsEnable' (to control up/down traps). The default value for this object is disabled state. No notification is sent in disabled state. Thus, the SET operation is also supported for this MIB object in order to control the emission of traps.

BFD Session Up/Down Traps

BFD has two traps, one for notifying that the session moved to the UP state, and the other trap for notifying that the session moved to DOWN state. To reduce the number of traps sent to NMS, a single trap is generated to combine the status changes of multiple sessions if the sessions have contiguous session IDs and multiple sessions move to either the UP or DOWN state in the same window of time. However, status changes of different types (UP & DOWN), will not be mixed in single trap. The window of time to combine the traps can be configured using the CLI command `configure snmp traps batch-delay bfd`.

For example, if sessions with session IDs 1, 2, 3, 4, and 5 are moving to the UP state in the same window of time, then a single trap is sent with low range index 1 and high

range index 5. As a second example, after all sessions moved to the UP state, session ID 2 goes DOWN and comes back UP before generating the first trap. In this case also, the first trap which is the UP trap, is set to include all sessions. Then, the second trap would be the DOWN trap for session ID 2, and finally the third trap would be the UP trap again for session ID 2. Thus, events are not missed or reordered.

NMS relates traps to sessions using only the session index which is provided in traps. It is necessary that the session index does not change until NMS retrieves session details via GET requests. To achieve this, the session will be retained for fifteen minutes after deletion is initiated by the BFD client (control protocol). During this period transmission and reception of BFD control packets will be stopped. If BFD protection is requested for the same destination again within this period, the same session index is reused. With this change, NMS can also have good history of the session to a particular destination.

Configuring SNMP Traps for BFD

To enable snmp traps for bfd:

```
enable snmp traps bfd {session-down | session-up}
```

To disable snmp traps for bfd:

```
disable snmp traps bfd {session-down | session-up}
```

To configure batch delay for sending the traps:

```
configure snmp traps batch-delay bfd {none | delay}
```

To display the configuration:

```
show snmp traps bfd
```

Note
SNMP (Simple Network Management Protocol) traps for BFD are disabled by default for both session-down and session-up.

Configuration Example

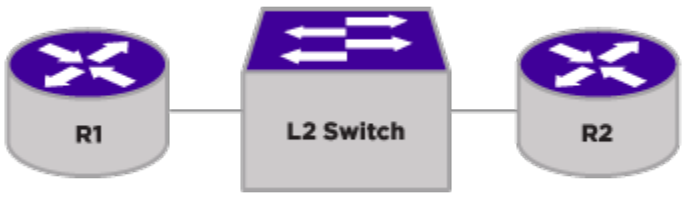


Figure 167: BFD Configuration Example

Consider the network segment like above, wherein two routers R1 and R2 are connected via an L2 switch. Following is the list of commands to configure BFD protection for static routes.

Router R1:

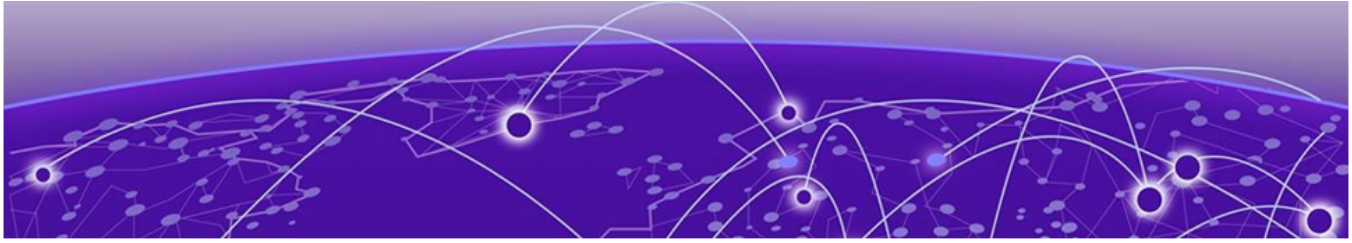
1. Create vlan and configure IP address.

   ```
   create vlan v1 tag 100

   configure vlan v1 add port 2 tagged

   configure vlan v1 ipaddress 10.0.0.1/24 2
   ```

2. Create BFD session to the next-hop which is being monitored.

   ```
   enable iproute bfd 10.0.0.2 vr VR-Default
   ```

3. Configure `iproute add <destination network> <gateway> bfd`



ONEPolicy

ONEPolicy Overview on page 1318
Policy Roles on page 1340
Classification Rules on page 1343
Authentication and ONEPolicy on page 1360
Configuring Policy on page 1366
ONEPolicy Configuration Examples on page 1379

ONEPolicy provides for the configuration of role-based profiles for securing and provisioning network resources based upon the role the user or device plays within the enterprise. By first defining the user or device role, network resources can be granularly tailored to a specific user, system, service, or port-based context by configuring and assigning rules to the policy role. A policy role can be configured for any combination of Class of Service, *VLAN (Virtual LAN)* assignment, or default behavior based upon L2, L3, and L4 packet fields. Hybrid authentication allows either policy or dynamic VLAN assignment, or both, to be applied through *RADIUS (Remote Authentication Dial In User Service)* authorization.

Note
The software only allows policy to be enabled if all the devices in the stack support policy. At the time of configuration the device will provision the lowest common denominator of functionality. If a device attempts to join the stack after policy is enabled, it must be able to support the existing level of functionality or it will not be allowed to participate in policy. For more detailed information about lowest common denominator, see Policy and Lowest Common Denominator Stacking on page 1336.

ONEPolicy Overview

The three primary benefits of using policy in your network are provisioning and control of network resources, security, and centralized operational efficiency. Policy provides for the provisioning and control of network resources by creating policy roles that allow you to determine network provisioning and control at the appropriate network layer, for a given user or device. With a role defined, rules can be created based upon up to 15 traffic classification types for traffic drop or forwarding. A *CoS (Class of Service)* can be associated with each role for purposes of setting priority, forwarding queue, rate limiting, and rate shaping.

Security can be enhanced by allowing only intended users and devices access to network protocols and capabilities. Some examples are:

- Ensuring that only approved stations can use *SNMP (Simple Network Management Protocol)*, preventing unauthorized stations from viewing, reading, and writing network management information
- Preventing edge clients from attaching network services that are appropriately restricted to data centers and managed by the enterprise IT organization such as *DHCP (Dynamic Host Configuration Protocol)* and DNS services
- Identifying and restricting routing to legitimate routing IP addresses to prevent DoS, spoofing, data integrity and other routing related security issues
- Ensuring that FTP/TFTP file transfers and firmware upgrades only originate from authorized file and configuration management servers
- Preventing clients from using legacy protocols

Note
Any configurations which require the use of the first stage ACL/VLAN processor, do not operate when OnePolicy is enabled. This includes, but is not limited to, certain MPLS, PSTag, VXLAN, and OAM/CFM configurations.

Note
Configuration changes on existing policy mux entries (changing the policy profile for a convergence endpoint to 0 or a different value, disabling *LLDP (Link Layer Discovery Protocol)* or CDP, etc.) do not take effect until re-authorization. As a result, existing CEP connections remain active and *FDB (forwarding database)* is still learned on policy profile even though CDP/LLDP neighbor times out and `show cdp neighbor {detail}` and `show lldp neighbors` is empty. You can force re-authorization by clearing a CEP connection: `configure policy convergence-endpoint clear ports [port_list | all]`.

Note
IDM and ONEPolicy are not supported together and it is not recommended to enable both, since handling rule/role-based actions is not supported, except to support Kerberos Authentication with NAC as a RADIUS server and can be used in conjunction with IDM XML event triggers.

Note
In ONEPolicy mode, when enabling NetLogin web-based, the following warning message appears when the port is not part of any default VLAN:
`WARNING: The following netlogin enabled ports 1 are not part of any VLAN. The port has to be part of some VLAN for Web-Based netlogin to work.`
For NetLogin web to work, the port must be part of a default VLAN.

Note
Restarting the NetLogin process is not supported when policy is enabled. Doing so results in indeterminate behavior.

Note

If Convergence End Point (CEP) (see Convergence End Point (CEP) Detection on page 1321) is configured and you have multiple authentication types configured, failure of a higher priority authentication results in the lower priority authentication being used.

```
# show netlogin session
Multiple authentication session entries
---------------------------------------
Port            : 3:1         Station address   : bc:f1:f2:b4:e7:5e
Auth status     : failed      Last attempt      : Fri Nov  4 13:39:34 2016
Agent type      : dot1x       Session applied   : false
Server type     : radius      VLAN-Tunnel-Attr  : None
Policy index    : 0           Policy name       : No Policy applied
Session timeout : 0           Session duration  : 0:00:00
Idle timeout    : 300         Idle time         : 0:00:00
Auth-Override   : enabled     Termination time: Not Terminated

Port            : 3:1         Station address   : bc:f1:f2:b4:e7:5e
Auth status     : success     Last attempt      : Fri Nov  4 13:38:49 2016
Agent type      : cep         Session applied   : true
Server type     : local       VLAN-Tunnel-Attr  : None
Policy index    : 1           Policy name       : Tes1 (active)
Session timeout : 0           Session duration  : 0:04:16
Idle timeout    : 300         Idle time         : 0:00:00
Auth-Override   : enabled     Termination time: Not Terminated

# show policy convergence-endpoint connections ports all

Convergence End Point Connection Info for port 3:1
Endpoint Type    cisco
Policy Index     1
Discovery Time   Fri Nov  4 13:38:49 2016
Firmware Version
Address Type     1
Endpoint IP
Endpoint MAC     bc:f1:f2:b4:e7:5e
```

Implementing Policy

To implement policy:

1. Identify the roles of users and devices in your organization that access the network.
2. Create a policy role for each identified user role (see Policy Roles on page 1340 and Table 98 on page 1366).
3. Associate classification rules and administrative profiles with each policy role (see Classification Rules on page 1343 and Table 99 on page 1368).
4. Optionally, configure a class of service and associate it directly with the policy role or through a classification rule (see Assigning a Class of Service to Policy Role on page 1342, Classification Rules on page 1343, and Table 98 on page 1366).
5. Optionally, enable hybrid authentication, which allows *RADIUS* filter-ID and tunnel attributes to be used to dynamically assign policy roles and VLANs to authenticating users (see Applying Policy Using Hybrid Authentication Mode on page 1364).

6. Optionally, set device response to invalid policy (see Device Response to Invalid Policy on page 1326).
7. Optionally, set captive portal to use HTTP redirection to force a client's web browser to be redirected to a particular administrative web page for authentication purposes (user login and password), payment (for example, at an airport hotspot), or use-policy enforcement (installing necessary software, agreeing to terms of service (TOS), etc.) (see Captive Portal Redirection and Table 102 on page 1372).

ONEPolicy Features

Convergence End Point (CEP) Detection

Convergence End Point (CEP) is a mechanism to detect remote IP telephony or video devices on a port and dynamically apply a policy based on the type of CEP device discovered. CEP is only active when ONEPolicy is enabled and configured on the switch. When a CEP device is detected on a port, the configured policy for that device is applied to the user on that port. The switch detects a CEP by inspecting devices on CDP- and *LLDP*-configured ports. CEP interacts with LLDP, CDP, and ONEPolicy through callbacks and/or inter-process messaging to initiate detection and apply policy.

LLDP and CDP

LLDP and CDP are required to call into CEP to add detected IP telephony devices with the following data:

- MAC—MAC address of detected device
- Port—port origin of detection
- Type—the type of detected device (Cisco, LLDP)
- inetAddr—IP address of end point device
- inetType—IP address type of end point device (IPv4, IPv6)
- inetLen—length of inetAddr

ONEPolicy

- CEP initializes with, and exists in, the ONEPolicy process and calls directly to modify policy muxRule entries.
- CEP leverages ONEPolicy port add/del/mod callback integration, satisfying CEP platform port requirements.

For information about configuring CEP detection, see Table 101 on page 1371.

Note

When both CEP and *NetLogin* are enabled on the same port, the policy profile name is "active" for both CEP and NetLogin sessions with session applied as "false" for CEP and "true" for NetLogin. If NetLogin authentication is successful, the session applied is false for CEP and true for NetLogin. NetLogin takes higher precedence than the CEP profile.

Learning of CEP entries depends on the LLDP/CDP update on active ports and on disabling and enabling CEP. New entries are learned only after receiving new LLDP/CDP information, and not from existing LLDP neighbors and CDP neighbors table.

For example:

```
# configure policy convergence-endpoint disable

# show fd
Mac                     Vlan       Age  Flags           Port / Virtual Port List
------------------------------------------------------------------------------

Flags : d - Dynamic, s - Static, p - Permanent, n - NetLogin, m - MAC, i - IP,
        x - IPX, l - lockdown MAC, L - lockdown-timeout MAC, M- Mirror, B - Egress
Blackhole,
        b - Ingress Blackhole, v - MAC-Based VLAN, P - Private VLAN, T - VLAN translation,
        D - drop packet, h - Hardware Aging, o - IEEE 802.1ah Backbone MAC,
        S - Software Controlled Deletion, r - MSRP,
        X - VXLAN, Z - OpenFlow

Total: 0 Static: 0  Perm: 0  Dyn: 0  Dropped: 0  Locked: 0  Locked with Timeout: 0
FDB Aging time: 300
* (Software Update Required) Slot-2 Stack.34 # sh lldp neighbors

        Neighbor           Neighbor                            Neighbor
Port    Chassis ID         Port ID            TTL     Age      System Name
==============================================================================
1:11    00:04:96:99:4F:EB  11                 120     3        switch_model
2:18    00:04:96:97:E9:EB  18                 120     4        switch_model
2:22    00:04:96:9A:5D:1C  17                 120     7        switch_model
2:23    00:04:96:52:E8:A4  15                 120     19       Not-Advertised
3:1     (5.1)10.120.93.33  70:38:EE:D0:91:6E  120     3        AVXD0916E
3:17    (5.1)10.127.6.192  D867D9E70736:P1    180     9
SEPd867d9e70736.extremenetworks.com
3:24    (5.1)0.0.0.0       BCF1F2B4E75E:P1    180     45       SEPBCF1F2B4E75E

# show cdp neighbor
Device Id            Local        Hold   Capability   Platform       Port Id
                     Interface    Time
------------------------------------------------------------------------------
00:04:96:99:4F:EB    1:11         130    RT   I       switch_model> Slot:  1, P>
00:04:96:9A:5D:1C    2:22         127     T           switch_model> Slot:  1, P>
SEPd867d9e70736      3:17         162        H        Cisco IP Phon> Port 1
SEPBCF1F2B4E75E      3:24         132        H        Cisco IP Phon> Port 1

# configure policy convergence-endpoint enable
Only after new LLDP/CDP packet is received CEP will be detected.
# show lldp neighbors
        Neighbor           Neighbor                            Neighbor
Port    Chassis ID         Port ID            TTL     Age      System Name
==============================================================================
1:11    00:04:96:99:4F:EB  11                 120     27       switch_model
2:18    00:04:96:97:E9:EB  18                 120     28       switch_model
2:22    00:04:96:9A:5D:1C  17                 120     1        switch_model
2:23    00:04:96:52:E8:A4  15                 120     13       Not-Advertised
3:1     (5.1)10.120.93.33  70:38:EE:D0:91:6E  120     27       AVXD0916E
3:17    (5.1)10.127.6.192  D867D9E70736:P1    180     3
SEPd867d9e70736.extremenetworks.com
3:24    (5.1)0.0.0.0       BCF1F2B4E75E:P1    180     39       SEPBCF1F2B4E75E
==============================================================================
NOTE: The Chassis ID and/or Port ID might be truncated to fit the screen.
```

```
# show fd
Mac                     Vlan       Age  Flags           Port / Virtual Port List
-------------------------------------------------------------------------------
00:04:96:99:4f:eb SYS_VLAN_1000(1000) 0000  dhm           1:11
70:38:ee:d0:91:6e SYS_VLAN_2000(2000) 0000 ndhm     v     3:1
bc:f1:f2:b4:e7:5e SYS_VLAN_1000(1000) 0041 nd m     v     3:24
d8:67:d9:e7:07:36 SYS_VLAN_1000(1000) 0000 n
```

Note

After CEP devices are mapped to a profile, changing the index value to "0" or to some other policy profile name, the existing CEP connections are still be mapped to the old profile that was configured initially when the CEP devices were detected. To force a refresh of existing detected devices, disable, and then enable, CEP (see `configure policy convergence-endpoint [` **`enable`** `|` **`disable`**`]`) or disable, and then enable, the port(s) (see `disable port [`*`port_list`* `|` **`all`**`]` and `enable port [`*`port_list`* `|` **`all`**`]`).

Captive Portal Redirection

Captive Portal Redirection uses HTTP redirection to force a client's web browser to be redirected to a particular administrative web page. You can use this feature for such web-based purposes as:

- Authentication—request a user login and password
- Payment—for example, at an airport hotspot
- Use-policy enforcement—require installing necessary software, agreeing to terms of service (TOS), etc.

Captive Portal Redirection is an extension of the ONEPolicy feature. You can configure policy roles that can force redirection of HTTP traffic by specifying a web redirection class index that associates it with up to two redirection servers. The HTTP traffic to potentially redirect is identified based on a destination captive portal server absolute URL address containing an IPv4 address, TCP port, and path. For traffic that is placed into one of these policy roles (through authentication or policy admin-profile rules) actions are taken based upon the contents of the policy profile.

If the incoming traffic is on the configured L4 port and is not destined for the configured captive portal server IP, the switch causes an HTTP redirect message (code 307) to be sent back to the client. If the incoming traffic is destined for the configured captive portal server IP, or it is not on one of the configured listening L4 ports, the traffic is handled according to the rest of the policy role configuration.

Configuring this feature occurs through the etsysPolicyProfileMIB and the ONEPolicy command set. There are two tables in the MIB, one that allows configuration of the listening ports and one that allows configuration of the captive portal servers. You can configure up to three ports on which ONEPolicy listens for client traffic that is (potentially) subject to HTTP redirection. A URL that explicitly identifies the server by an IPv4 address, TCP port, and path is configured along with the ports on which the Captive Portal feature listens for client traffic (for example: `configure policy captive-portal listening 80,8080`).

You can configure up to ten web-redirect groups of two captive portal servers. They can be used to redirect traffic in different roles to different servers. These web-redirect groups are identified by associating a web redirection class index with the server ID. The policy roles used for captive portal redirection each have a non-zero web redirection class index configured (for example, `configure policy profile 1 web-redirect 5`). The default captive portal web redirection class index for any given role (profile) is 0, or unset. To enable captive portal, there must be a role defined that has a valid captive portal web redirection class index with a value between 1–10.

In addition to the captive portal configuration, this policy role should also have rules to handle the traffic that is not handled by the captive portal web redirection.

Note

When there are two servers configured in a web-redirect, the switch uses the following algorithm to pick which server to use for redirection:

((Last byte of the client's source MAC address)%(numServers)) + 1

For example, (mac = 00:00:00:00:00:03) and where numServers is 2. (0x03%2) + 1 = 2 (This MAC uses server 2.)

Note

Captive portal does not work with tagged frames.

Note

HTTPS redirection is not supported in Captive Portal redirection.

Redirection to the portal does not happen if the user (guest-user or unauthenticated user) tries to connect to an HTTPS website. This feature only handles HTTP traffic. However, you can configure redirection to a captive portal URL based on HTTPS.

If not specified to do otherwise, ONEPolicy programs its captive portal-related rules outside of the reserved ACL rule space for ONEPolicy. This results in additional ACL slice usage. Starting with ExtremeXOS 30.4, you can specify that these rules are programmed within the already reserved ACL rule space at the expense of IPv4 rule capacity (see Table 98 on page 1366).

For information about configuring Captive Portal Redirection, see Table 102 on page 1372.

For an example Captive Portal Redirection configuration, see Captive Portal Redirection Example on page 1389.

Dynamic Captive Portal Redirection Using RADIUS VSA

Version 33.5.1 adds support for dynamic Captive Portal redirection through RADIUS Vendor-Specific Attributes (Extreme-URL-Redirect with value 234). This enhancement enables Network Access Control (NAC) servers to dynamically specify web redirect servers during user authentication, replacing the previous requirement for static switch configuration. The new RADIUS VSA accepts either IPV4 address or FQDN for the

redirect server. This feature addresses password expiration scenarios by allowing clients to be redirected to self-service password renewal portals.

The following show commands are enhanced to support this feature:

show netlogin session

show policy captive-portal

Example

```
# show netlogin session
Multiple authentication session entries
---------------------------------------

Port            : 25          Station address   : 00:0a:01:00:00:02
Auth status     : success     Last attempt      : Mon Sep 22 18:17:20 2025
Agent type      : mac         Session applied   : true
Server type     : radius      VLAN-Tunnel-Attr  : None
Policy index    : 2           Policy name       : profile2 (active)
Session timeout : 40          Session duration  : 0:00:01
Idle timeout    : 50          Idle time         : 0:00:00
Auth-Override   : disabled    Termination time  : Not Terminated
Dynamic Web-redirect : http://2.2.2.2:8012/
* (CIT_33.5.0.259) 5320-24P-8XE-SwitchEngine.5 # sh policy captive-portal web-redirect
00:0a:01:00:00:02

Dynamic Web-redirect  : http://2.2.2.2:8012/
```

Limitations

- Only one `Extreme-URL-Redirect` VSA per user session is processed
- Maximum 20 unique URL redirects supported simultaneously
- HTTP traffic only (HTTPS traffic is not redirected, though redirect URL can be HTTPS)
- Extreme-URL-Redirect VSA in COA messages are ignored
- The `Extreme-URL-Redirect` RADIUS VSA operates independently of the `configure policy maptable response` setting. If a user is assigned a policy, the redirect URL is installed using the policy profile index provided by RADIUS. If no policy is assigned, the session is administratively reset. The VSA does not honor the maptable response configuration, and URL redirection will function only when a valid policy is associated with the user.
- The command **configure policy captive-portal rule-use reserved** is intended to install captive portal redirect rules in reserved policy slices. However, this behavior does not apply to ACLs installed using URL redirect RADIUS attributes. These ACLs are currently placed outside the reserved slice, regardless of the configuration.
- Only the first valid attribute in the Access-Accept message is accepted; any subsequent attributes are ignored.
- The Extreme-URL-Redirect value can be either an IPv4 address or a fully qualified domain name (FQDN). An FQDN is accepted only if it can be resolved. This behavior is consistent with CLI-based configuration.
- Only one authentication agent type—either MAC or 802.1X—can use the Extreme-URL-Redirect VSA. The attribute from the first successfully authenticated agent

is processed; any subsequent attributes are ignored. Changing the authentication order after the initial authentication has no effect, and the URL from the first authenticated method will be used for the client.

- It is recommended to set the rule-use option to "unreserved" if both CLI-configured and RADIUS-provided URL information are expected to be processed by the system.

VLAN/VXLAN to Policy Mapping

VLAN-to-Policy mapping manually configures VLAN-to-policy associations that create a policy maptable entry between the specified VLAN and the specified policy role. When an incoming tagged VLAN packet is seen by the switch, a lookup of the policy maptable determines whether a VLAN-to-policy mapping exists. This feature can be used at the distribution layer in environments where non-policy-capable edge switches are deployed and you cannot apply Extreme policy at the edge. Tagged frames received at the distribution layer interface for a VLAN with an entry in the policy maptable have the associated policy applied to the frame.

To create a policy maptable entry associating a policy to a VLAN, use the following command specifying a single VLAN ID or range of IDs and the policy profile-index:

```
configure policy maptable [response [tunnel | policy | both] | vlan_list
profile_index]
```

Starting with ExtremeXOS 22.5, you can map *VXLAN (Virtual Extensible LAN)*s by applying the VXLAN identifier (VNI) to a profile using the command, where NSI = VNI:

```
configure policy profile profile_index {name name} {pvid pvid}
{pvid-status pvid_status} {cos cos} {cos-status cos_status} {egress-
vlans egress_vlan_list}{forbidden-vlans forbidden_vlans} {untagged-vlans
untagged_vlans} {append | clear} {tci-overwrite tci_overwrite}
{precedence [precedence | default]} {auth-override auth_override} {nsi
[nsi | none]} {web-redirect web_redir_index} {access-list [unassigned |
list_name | list_name_placeholder]}
```

You can then create the maptable entry, using the previous maptable command.

Device Response to Invalid Policy

The action that the device should take when asked to apply an invalid or unknown policy can be specified. The available actions are:

- Ignore the result and search for the next policy assignment rule. If all rules are missed, the default policy is applied.
- Block traffic.
- Forward traffic as if no policy has been assigned using 802.1D/Q rules.

Use the `configure policy invalid action` command to specify a default action to take when asked to apply an invalid or unknown policy.

Policy-Based Mirrors

You can apply mirrors to policy profile rules by using a "control group" mirror referenced by a unique control index number. These control group mirrors are

etsysMirrorDestinationControlEntry entries in the ENTERASYS-MIRROR-CONFIG-MIB (Mirror MIB) (see ENTERASYS-MIRROR-CONFIG-MIB on page 1894). A Mirror MIB instance (designated by a control index) can be associated with up to four "physical" mirrors, each being one destination port (or tunnel).

For information about configuring policy-based mirrors, see Table 103 on page 1373.

Limitations

- Mirrors with multiple destination ports are not supported.
- A maximum of four concurrently enabled mirrors is supported.
- A maximum of four control group (Mirror MIB) instances can be created using the CLI.
- Mirrors on admin profile rules is not supported.
- Mirroring cannot be enabled on LAG and MLAG ports.
- Mirroring index cannot be assigned for default mirrors.

Rule Trap and Syslog for ONEPolicy

You can be notified when a policy rule is used by enabling “Syslog” or “trap” actions:

- The Syslog action sends an entry in the log after first rule use with the EMS tag “Policy.LogRuleHit”.
- The trap action sends an etsysPolicyRulePortHitNotification SNMP trap to the configured trap receiver.

By default, the Syslog and trap actions only occur when the rule is first used. However, for the Syslog action, you can configure the system to send messages every time the rule is used at a maximum rate of once every five seconds.

Additionally, a rule counter tracks the number of times a packet triggers a rule. By default, the counter is enabled for all rules.You can set an interval for how often this counter is cleared, or manually clear it at any time.

For information about configuring rule trap/Syslog, and the rule counter, see Table 104 on page 1374.

Limitations

Not supported on admin profile rules.

VCAP Partitioning

ExtremeXOS allows multiple applications (Layer 7 policy, user-based dynamic ACLs) to share a pool of VCAP slices. This area is called the “shared pool.” The TCI overwrite-enabled case is configurable to use its own pool of VCAP slices and the “shared pool” is allocated for the other applications. VCAP slices not allocated for policy usage are available for general system use.

TCI overwrite-enabled profiles use their own resource pool of slices (0–4) for VLAN assignment (translation). Allocating less TCI overwrite-enabled VCAP space means fewer rules available for users to have their VLAN tags overwritten.

Supported Platforms

All ExtremeSwitching Universal platforms.

Layer 7 Policy/Application Signature

Policy rules are used to assign incoming traffic to a specific policy profile. Layer 7 policy/application signature provides an additional traffic classification capability. This layer 7 classification is accomplished by the snooping of DNS packets for pre-defined traffic application signatures. The snooping feature is activated automatically when policy is enabled and an application signature (AS) rule is created. It is deactivated if all AS rules are removed or if policy is disabled. Layer 7 policy uses shared look-up stage TCAM resources that must be configured first (see VCAP Partitioning on page 1327). The number of signature rules is dependent upon the availability of this shared space.

Dynamic policy rules are created in hardware using DNS-discovered destination IP addresses for the host name. The IP addresses ultimately map the traffic to a particular application traffic signature pattern. IP addresses and their associated dynamic rules are timed out per the TTL of the snooped IP address. The application traffic signatures are defined in groups of one or more signature names, where each signature name consists of one or more signature patterns.

Layer 7 policy is based on the use of the enterasys-application-signature-mib (see ENTERASYS-APPLICATION-SIGNATURE-MIB on page 1872).

Supported Platforms

All ExtremeSwitching Universal platforms.

Limitations

- V6 is not supported.
- Rule creation and deletion are controlled by a time-to-live (TTL) 5 second timer.
- TTL value is configurable in terms of 1 min., 5 mins., and 10 mins.
- When VCAP space runs out, any IP rule not already installed in hardware is not created. New attempts to install previously uninstalled rules occur after the 5 second TTL timer interval and are dependent upon the space made available by other rules that have timed out.

Signatures

Starting with ExtremeXOS 30.4, policy rules have an additional traffic classification of application signature group and name. There is a fixed, defined set of signature groups available. Each group has a subset of pre-defined/built-in signature names; these can be enhanced with additional ones that can be created by commands or MIB. Additionally, each application signature name is defined by one or more signature patterns.

The signatures are arranged into two hierarchical levels, group and name.

Group examples: Advertising, Storage, Education

Names examples: Etrade, Skype, AOL, Facebook, Twitter, Google

Example Signature Groups

Built-in signatures:

- Social Networking
  - Facebook
    - facebook.com
    - facebook.net
- Search Engines
  - Google
    - google.co
    - google.ca
    - google.it
    - google.fr

User-defined signatures:

- News and Information
  - Flipboard
    - flipboard.com
- E-commerce
  - Warehouse
    - bjs.com
    - costco.com

Pattern examples:

- wireshark.org
- mysql.com

ACL Style Policy

Traditional ONEPolicy architecture uses a hierarchical approach to rule precedence where rule type dictates precedence. In addition, rule look-ups occur per role, per action type. This means, for example, that triggering a forward/drop rule without an explicit Class of Service (CoS) action results in applying the forward/drop action, and then continuing searching until a rule with CoS action matches. This hierarchical approach is implemented in hardware by maintaining one list for forward/drop actions, and one list for CoS actions. This implementation often results in underused resources, because not every rule has both forward/drop and CoS actions.

With ACL Style Policy, a mode of operation with a single ordered list per role is maintained. Rule look-ups occur in the provided ACL order per role. A match applies all actions specified, and the search stops. This approach can potentially double the advertised scale of classification rules as compared to the traditional model. It also provides a more standard approach to policy classification rules.

Default policy rule-model is hierarchical, unless you are upgrading from ExtremeXOS 30.5 with a saved configuration that is set to access list.

For information about configuring ACL Style Policy, see Table 106 on page 1377.

Limitations

SNMP for configuration of ACL Style Policy classification rules is not supported.

Policy access-list hardware installation errors may occur depending on the configured IFP slice mode, the type of IFP processor, and the combination of match criteria used. The `ipv4-ipv6-double-shared` slice mode attempts to program all policy access-lists into a single double-wide slice type. This allows installation of match fields such as IPv4/IPv6 destination prefixes, layer 4 destination ports (including ranges), and Ethernet type. However, if additional match criteria—such as IPv4 source address, layer 4 source ports, TTL, or TOS—are required, the `ipv4-ipv6-double-separate` mode should be used to ensure successful hardware programming.

Similar limitations apply to the `ipv4-single-ipv6-double` slice mode. In this configuration, IPv4 access-lists are programmed in single-wide slices, while IPv6 access-lists use double-wide slices—only on platforms that support double-wide. On platforms that do not support double-wide slices, IPv6 access-lists will also be programmed in single-wide slices, which restricts the available match criteria.

Platforms that do not support double-wide slices:

- 4120
- 5120
- 5320-16P-2MXT

Traditional Versus ACL Style Policy Classification Rule Scaling

The following table compares the overall classification rule scale between "traditional" and ACL Style policy:

Table 87: Traditional versus ACL Style policy classification rule scaling

Switch Model(s)	Traditional	ACL Style**
4120	Default: 184 Less System ACL: 184	Default: 440 Less System ACL: N/A
4220	Default: 440 Less System ACL: 440	Default: 952 Less System ACL: N/A
5120	Default: 184 Less System ACL: 184	Default: 440 Less System ACL: N/A
5320-48T	Default: 1,976 Less System ACL: 1,976	Default: 4,024 Less System ACL: N/A
5320-24T-4X-XT	Default: 440 Less System ACL: 440	Default: 952 Less System ACL: N/A
5320-24T-24S-4XE-XT	Default: 1,976 Less System ACL: 1,976	Default: 4,024 Less System ACL: N/A
5320-16P*	Default: 1,976 Less System ACL: 1,976	Default: 4,024 Less System ACL: N/A

Table 87: Traditional versus ACL Style policy classification rule scaling (continued)

Switch Model(s)	Traditional	ACL Style**
5420M	Default: 4,024 Less System ACL: 4.024	Default: 8,120 Less System ACL: N/A
5420F	Default: 1,976 Less System ACL: 1,976	Default: 4,024 Less System ACL: N/A
5520	Default: 4,024 Less System ACL: 4.024	Default: 8,120 Less System ACL: N/A
5720-MW	Default: 6,072 Less System ACL: 6,072	Default: 12,216 Less System ACL: N/A
5720-MXW	Default: 8,120 Less System ACL: 8,120	Default: 16,312 Less System ACL: N/A
7520-48Y-8C	Default: 1,976 Less System ACL: 1,976	Default: 3,512 Less System ACL: N/A
7520-48XT-6C	Default: 1,976 Less System ACL: 1,976	Default: 3,512 Less System ACL: N/A
7720-32C	Default: 1,976 Less System ACL: 1,976	Default: 3,512 Less System ACL: N/A

* - 5320-16P models do not support MAC and IPv6 rules in traditional mode.

** - Applies to role-based DACLs as well as static ACLs created via the Policy CLI, which are also associated with a profile. User-based DACLs may achieve lower ACL Style numbers.

ACL Style Classification Rule Architecture

- Single ordered list per role.
- Rule look-ups done in provided ACL order.
- Rule look-up match applies all actions, and then the search stops.
- 5-tuple rule match support.
- Layer 4 port range.
- Rule types: application, ip6dest, ipsourcesocket, ipdestsocket, ipfrag, udpsourceportIP, udpdestportIP, tcpsourceportIP, tcpdestportIP, icmptype, icmp6type, ipttl, iptos, ipproto, ether.
- Rule actions: permit/deny, CoS (prio, tos, rate-limit, shape, etc), mirror, Syslog, and count.

RESTful API Support

ACL Style Policy implements a new RESTful API for configuration of classification rules.

Note
RESTful API is not supported for Dynamically-created ACLs via RADIUS and COA.

Dynamic ACLs

Dynamic *ACL (Access Control List)*s are created using the CLI. They use a similar syntax and can accomplish the same actions as single rule entries used in ACL policy files. More than one dynamic ACL can be applied to an interface, and the precedence among the dynamic ACLs can be configured. By default, the priority among dynamic ACLs is established by the order in which they are configured.

Note
Dynamic ACLs have a higher precedence than ACLs applied using a policy file.

The steps involved in using a dynamic ACL on an interface are:

- Creating the Dynamic ACL Rule
- Configuring the ACL Rule on the Interface.
- Configuring ACL Priority
- Network-Zone Support in ACLs

Role-based ACLs

Beginning with Release 32.1, the dynamic ACL rule(s) can be user-based or role-based. User-based rules are treated as higher priority than any other statically provisioned rules. Policy roles and the DACLs associated with them are dynamically created as needed based on the incoming RADIUS Filter-Id attribute. This attribute is automatically deleted when the last authenticated user associated with the role is removed.

When a set of role-based rules is installed for a given role or profile, they cannot be changed until that role is no longer in use. Role-based rules are shared by any other user who authenticates to the same role or profile. While both user based and role-based DACLs can be used on the device at the same time, a mix of user based and role-based DACLs are not permitted for a given user.

A role-based operation has a type 'r' and requires a preceding add operation (a,r). Each role requires a profile pre-configured with a unique name and access-list configuration.

A role-based with create operation has a type 'c' and also requires a preceding add operation (a,rc). The role or profile is dynamically created if it does not already exist. If created dynamically, the role or profile will be deleted when no longer in use.

A delete-all operation has a type of 'da' and no match, action, or index fields are permitted. When used, the delete-all must be the first entry in the list. When present, this operation removes all existing rules associated with the user or role. Neither the action field nor the index field is permitted and will be ignored if present.

The following match conditions can be used with role-based ACLs:

- `ipv4src` *`ipv4source/mask-length`*
- `ipv4dst` *`ipv4dest/mask-length`*
- `ipproto` *`ipproto`* (TCP, UDP, ICMP, or protocol number)(ICMP and protocol number as of Release 32.1)

- `l4srcport` *`l4sourceport-l4sourceportend/mask-length`* (requires ipproto; range is role-based only and no mask)
- `l4dstport` *`l4destport-l4destportend/mask-length`* (requires ipproto; range is role-based only and no mask)
- `ether` (role-based only)
- `any` (as of version 32.1)
- `ip6dest` (as of version 33.4.1

Supported Platforms

All Universal platforms.

Limitations

- DNS is not supported on Extended Edge switches with Controlling Bridges on 5420 and 5520 Series switches.
- ACL style policy must be selected.
- Only a subset of the existing policy rules is allowed.
- SNMP is not supported.
- Not all rule types can be configured at the same time. For example, L4 port range rules with ether and other IP and masked L4 ports rules may not install correctly. This is the same limitation that applies to ACL Style Policy.

Access List Mode Enhancements for Double Wide Slices

Version 33.4.1 adds additional support for specifying single, single-double, double-only, double-separate, or double-shared slices for IPv4 and IPv6 rules. The default setting (ipv4-single-only) is equivalent to the existing behavior where only single-wide slices are used. This enhancement includes both statically configured and role-based downloadable ACLs.

The following command is added in support of this feature:

`configure policy slices ingress [`**`ipv4-single-only`** | **`ipv4-single-ipv6-double`** | **`ipv4-double-only`** | **`ipv4-ipv6-double-separate`** | **`ipv4-ipv6-double-shared`**`]`

Double-wide mode effectively decreases the ACL mode limit in half.

The following table lists Policy ACL scaling with double-side slices per platform:

Platform	Total IFP slices	Policy IFP Slices	IFP Slice Depth	Max single wide ACLs	Max Double Wide Slices	Max Double Wide Rules
5420	12	8	512	4024	5	2488
5320-16/24	8	4	1024	4024	2	1976
4220	8	4	256	952	2	440
5520	12	8	1024	8120	3	3000
5720-MW	12	8	1536	12216	2	3000

Platform	Total IFP slices	Policy IFP Slices	IFP Slice Depth	Max single wide ACLs	Max Double Wide Slices	Max Double Wide Rules
5720-MWX	12	8	2048	16312	5	10168
7520	12	7	512	3512	4	1976
7720	12	7	512	3512	4	1976

Some access-lists may not install in hardware, depending on how your system is configured. Things to note:

ipv4-ipv6-double-shared

- Installs both IPv4 and IPv6 rules in the same double-wide slice.
- Works well for **destination-based matches** like:
  - IPv4/IPv6 destination addresses
  - UDP destination ports (and ranges)
  - Ethernet type

These examples will install successfully:

```
create policy access-list acl1.ace1 matches ip6dest 2001::1 mask 128 udpdestportIP 50
actions cos 2 syslog forward mirror 1
create policy access-list acl1.ace2 matches ip6dest 2002::1 mask 128 udpdestportIP 51-100
actions cos 2 syslog forward mirror 1
create policy access-list acl1.ace3 matches ether 0x86dd actions drop

create policy access-list acl1.ace4 matches ipdestsocket 10.1.1.1:20 mask 48 actions cos
2 syslog forward mirror 1
create policy access-list acl1.ace5 matches ipdestsocket 10.1.1.2:21-30 mask 64 actions
cos 2 syslog forward mirror 1
create policy access-list acl1.ace6 matches ether 0x0800 actions drop
```

If you need to match on source IPs, source ports, TTL, or TOS, this mode may not work.

ipv4-ipv6-double-separate

- Installs IPv4 and IPv6 rules in separate slices.
- Use this mode if you need more complex match conditions (like source-based matches).

ipv4-single-ipv6-double

- Also separates IPv4 and IPv6 rules.
- IPv4 uses single-wide slices.
- IPv6 uses double-wide slices **if supported by the platform**.
- On platforms without double-wide support, IPv6 rules are also single-wide, so the same match limitations apply as in the shared mode.

Limitations

This feature has the following limitations:

- Policy must be disabled to change the ingress slice mode.
- The precedence of policy slices can't be changed.
- The setting only applies to access-list mode.
- ACL entries—static or role-based dynamic—cannot combine `ipv6dst` with either `ether` or Layer 4 source port (`l4sourceport`). These match conditions must be used in separate rules.
- Slice mode must support IPv6 to use ipv6dst match condition (CLI will give an error for statically configured and the rule will not install. A log message is generated and attempted with Dynamic ACLs).
- On platforms with limited ICAP slice availability (e.g., 4 total slices), enabling features such as MACsec, IPv6 multicast, and complex ACL match criteria (e.g., ethertype, L4 port ranges, IP addresses) may exhaust available slice resources. When `double-wide` slice mode is selected, each reserved slice consumes two physical slices. For example, if an IPv6 address is configured on a VLAN and a slice is reserved for IPv6 multicast, attempts to install additional ACLs may fail with errors such as:

  ```
  pibIPv4AclSliceConfigReserve(slot=1, unit=0, doubleWide=1, numSlices=1) failed
  ```

  Despite this, ACL rules will continue to install successfully until the first double-wide slice is filled. This behavior is platform-dependent and may vary based on configuration.
- When using the `ipv4-double-only` ingress slice mode, a failure may occur during Dynamic ACL (DACL) installation if an IPv6 VLAN is present and has already reserved a slice for IPv6 multicast traffic. This results in insufficient slice resources for IPv4 double-wide ACLs.

Supported Platforms

IPv4 double-only, IPv4/IPv6 double-separate, and double-shared are supported on 5320 (except 5320-16P-2MXT-2X), 5420, 4220, 5520, 5720, 7520, 7720 Series switches.

The following platforms support the `ipv4-single-ipv6-double` mode. However, the slices on these platforms are single-wide, which enables IPv6 support but with certain limitations. Due to the single-wide nature of IPv6 slices, only a limited set of match criteria can be installed in hardware. In general, destination-based rules—such as those matching on IPv4, IPv6, Ethernet, or Layer 4 source port—will install successfully. However, combining destination-based matches with source-based criteria may result in hardware installation failures:

- 5120
- 5320-16P-2MXT-2X

The following table lists old style ICAP (double-wide not supported, IPv6 supported):

Platform	Total IFP slices	Policy IFP Slices	IFP Slice Depth	Max total ACLs	Max Double Wide Slices	Max IPv4 or IPv6 ACLs in ipv4-single-ipv6-double mode (assume 1 slice for other type)
4120	8	4	128	512	NA	384
5120	8	4	128	512	NA	384

Policy and Lowest Common Denominator Stacking

When devices are stacked that have different capacities, we sink to the lowest common level of capacities and functionality when possible. If the stack already has an existing configuration that exceeds the new lower capacity policy will disallow the ports on the new unit to become policy enabled.

Note
In some cases policy capacities of a device have been determined to be so low that they are deigned non-useful for any practical policy deployment. In these cases the ports on the device are never allowed to become policy enabled.

The following algorithm is used to determine the lowest common denominator when stacking:

```
(A) More Capable device(s)
(B) Less Capable device(s)

- (A) is running and (B) is added

   If (A) is currently using less or equal resources to the maximum
   capacity (B) provides.
       - Policy will reduce the system maximum capacity
        to match (B).
   If (A) is currently using more resources than the maximum capacity (B)
provides.
        - Policy will not be enabled on the ports of (B).
        - A message will be printed that indicates the condition to the
          customer. If the resources being used by (A) are reduced to less or equal the
maximum capacity of (B)
        the ports on (B) will become policy enabled. A message will be printed that
indicates the condition to the customer and thereafter every
        5 minutes as the condition exists. If the resources being used by (A) are reduced
to less or equal the maximum capacity of (B)
        a message will be printed that indicates this slot can fully function with policy
pending a
        reboot of (B). This message will repeat every 5 minutes until (B) is
        rebooted.               - (B) is running and
        (A) is added      o The
        system maximum capacity will continue to match (B)
        - If (B) is removed (A) will revert to its higher capacity.                 -
(A) and (B) brought
        up at the same time.      o The
```

```
        system maximum capacity will be the lower capacity of (B)- If (B) is removed
        (A) will revert to its higher maximum capacity.
```

Standard and Enhanced Policy Considerations

This section itemizes additional policy considerations for the stackable and standalone platforms, and provides a table cross-referencing standard and enhanced policy capability and policy capability to traffic classification rules.

Not all stackable fixed switch platforms support policy. On some stackable and standalone fixed switch platforms policy support requires a purchased license. See the software release notes that come with your device for policy support.

Table 88 provides a listing of policy capabilities by standard and enhanced support level. Standard policy capabilities are further granulated based upon traffic classification support. See Table 89 on page 1338 for a cross-reference of traffic classification to policy capability support.

Table 88: Standard and Enhanced Policy Capability Cross-Reference

Policy Support Level	Policy Capability
Standard	• Dynamic PID Assign Rule – The ability to dynamically assign a policy based upon a traffic classification (macsource and port-string). See **Dynamic** in Table 89 on page 1338. • Admin PID Assign Rule – The ability to administratively assign a policy based upon a traffic classification (macsource and port-string). See **Admin** in Table 89 on page 1338. • *VLAN* Forwarding – The ability to assign a forwarding VLAN rule through the default profile/role PVID only. • Deny – The ability to assign a drop traffic rule. See **Drop** in Table 89 on page 1338. • Permit – The ability to assign a forward traffic rule. See **Forward** in Table 89 on page 1338. • *CoS* Assign Rule – The ability to assign a CoS rule. See **CoS** in Table 89 on page 1338. • Priority – The ability to assign traffic priority using a CoS assignment. See **CoS** in Table 89 on page 1338. • Longest Prefix Rules – The ability to always look at the highest bit mask for an exact traffic classification match.
Enhanced	• TCI Overwrite – The ability to overwrite user priority and other VLAN tag TCI field classification information. • Invalid Policy Action – The ability to set a drop, forward, or default-policy behavior based upon an invalid action.

Table 89 provides a cross-reference of standard ( ) and enhanced (X) policy capability to traffic classification rule.

Table 89: Policy Capability to Traffic Classification Rule Cross-Reference

Traffic Classification Rule	Dynamic	Admin	VLAN	CoS	Drop	Forward	Syslog	Trap	Mirror
MAC Source Address	X	X		X	X	X	X	X	X
MAC Destination Address				X	X	X	X	X	X
IPX Source Address							X	X	X
IPv6 Destination Address				X	X	X	X	X	X
IPX Destination Address							X	X	X
IPX Source Socket							X	X	X
IPX Destination Socket							X	X	X
IPX Transmission Control							X	X	X
IPX Type Field							X	X	X
IP Source Address				X	X	X	X	X	X
IP Destination Address				X	X	X	X	X	X
IP Fragmentation				X	X	X	X	X	X
UPD Port Source				X	X	X	X	X	X
UDP Port Destination				X	X	X	X	X	X
TCP Port Source				X	X	X	X	X	X
TCP Port Destination				X	X	X	X	X	X

Table 89: Policy Capability to Traffic Classification Rule Cross-Reference (continued)

Traffic Classification Rule	Dynamic	Admin	VLAN	CoS	Drop	Forward	Syslog	Trap	Mirror
ICMP (Internet Control Message Protocol) Packet Type							X	X	X
Time-To-Live (TTL)				X	X	X	X	X	X
IP Type of Service				X	X	X	X	X	X
IP Protocol				X	X	X	X	X	X
Ether II Packet Type				X	X	X	X	X	X
LLC DSAP/ SSAP/CTRL							X	X	X
VLAN Tag							X	X	X
TCI-Overwrite							X	X	X
Port String	X	X		X	X	X	X	X	X

Loop Prevention with Spanning Tree Protocol

See STP and ONEPolicy on page 1842.

Terms and Definitions

Table 90 lists terms and definitions used in this policy configuration discussion.

Table 90: Policy Configuration Terms and Definitions

Term	Definition
Administrative Profile	A logical container that assigns a traffic classification to a policy role.
CoS	A logical container for packet priority, queue, and forwarding treatment that determines how the firmware treats a packet as it transits the link.
Filter-ID	A string that is formatted in the *RADIUS* access-accept packet sent back from the authentication server to the switch during the authentication process. In the Extreme policy context, the string contains the name of the policy role to be applied to the authenticating user or device.

Table 90: Policy Configuration Terms and Definitions (continued)

Term	Definition
Hybrid Authentication	An authentication feature that allows the switch to use both the filter-ID and tunnel attributes in the RADIUS response message to determine how to treat the authenticating user.
Policy	A component of Secure Networks that provides for the configuration of a role based profile for the securing and provisioning of network resources based upon the function the user or device plays within the enterprise network.
Policy Maptable	A logical entity that can be configured to provide *VLAN* to policy role mappings.
Policy Profile/Role	A logical container for the rules that define a particular policy role.
Policy Rule	A logical container providing for the specification of policy behaviors associated with a policy role.
Role	The grouping of individual users or devices into a logical behavioral profile for the purpose of applying policy.
Rule Precedence	A numeric traffic classification value, associated with the policy role, the ordering of which on a precedence list determines the sequence in which classification rules are applied to a packet. **Note:** Rule precedence is fixed (not configurable) until ExtremeXOS 33.6.1 and later.
TCI Overwrite	A policy feature, when enabled in a policy role-based tci-overwrite only, allows for the overwrite of the current user priority and other classification information in the VLAN tag's TCI field.
Traffic Classification	A network element such as MAC or IP address, packet type, port, or VLAN used as the basis for identifying the traffic to which the policy will be applied.
Untagged and Tagged VLAN	Untagged VLAN frames are classified to the VLAN associated with the port it enters. Tagged VLAN frames are classified to the VLAN specified in the VLAN tag; the PVID is ignored.
VLAN Authorization	An aspect of RFC3580 that provides for the inclusion of the VLAN tunnel attribute in the RADIUS Access-Accept packet defining the base VLAN-ID to be applied to the authenticating user or device.
VLAN Egress List	A configured list of ports that a frame for this VLAN can exit.

Policy Roles

The following sections provide information about policy roles.

The Policy Role

A role, such as sales, admin, or engineering, is first identified and defined in the abstract as the basis for configuring a policy role. Once a role is defined, a policy role is configured and applied to the appropriate context using a set of rules that can control and prioritize various types of network traffic. The rules that make up a policy role contain both classification definitions and actions to be enforced when a classification is matched. Classifications include Layer 2, Layer 3, and Layer 4 packet fields. Policy actions that can be enforced include *VLAN* assignment, filtering, inbound rate limiting, outbound rate shaping, priority class mapping.

Defining a Policy Role

The policy role is a container that holds all aspects of policy configuration for a specific role. Policy roles are identified by a numeric profile-index value between 1 and the maximum number of roles supported on the platform. Policy roles are configured using the `configure policy profile` command. Policy configuration is either directly specified with the `configure policy profile` command or is associated with the role by specifying the *`profile-index`* value within the command syntax where the given policy option is configured. For example, when configuring a policy maptable entry using the `configure policy maptable` command (see VLAN/VXLAN to Policy Mapping on page 1326), the command syntax requires that you identify the policy role the maptable entry will be associated with, by specifying the profile-index value.

When modifying an existing policy role the default behavior is to replace the existing role with the new policy role configuration. Use the append option to limit the change to the existing policy role to the options specified in the entered command.

A policy role can also be identified by a text name of between 1 and 64 characters. This name value is used by the *RADIUS* filter-ID attribute to identify the policy role to be applied by the switch with a successful authentication.

Roles in a Secure Network

The capacity to define roles is directly derived from the ability of supported devices to isolate packet flows by inspecting Layer 2, Layer 3, and Layer 4 packet fields while maintaining line rate. This capability allows for the granular application of a policy to a:

- Specific user (MAC, IP address or interface)
- Group of users (masked MAC or IP address)
- System (IP address)
- Service (such as TCP or UDP)
- Port (physical or application)

Because users, devices, and applications are all identifiable within a flow, a network administrator has the capacity to define and control network access and usage by the actual role the user or device plays in the network. The nature of the security challenge, application access, or amount of network resource required by a given attached user or device, is very much dependent upon the “role” that user or device plays in the enterprise. Defining and applying each role assures that network access and resource

usage align with the security requirements, network capabilities, and legitimate user needs as defined by the network administrator.

Setting Default VLAN for Policy Role

A default *VLAN* can be configured for a policy role. A default VLAN will only be used when either a VLAN is not specifically assigned by a classification rule or all policy role classification rules are missed. To configure a default VLAN, enable pvid-status and specify the port VLAN to be used. pvid-status is disabled by default.

Note

ExtremeXOS supports the assignment of port VLAN IDs 1–4,094. Use of VLAN ID 4094 is supported by stackable and standalone devices. VLAN IDs 0 and 4095 cannot be assigned as port VLAN IDs, but 0 has a special meaning within a policy context and can be assigned to the PVID parameter. Within a policy context 0 specifies an explicit deny of all VLANs.

If policy profiles with PVID 0 and PVID 4095 are configured and for MAC authentication profile with PVID 0 is sent using Filter ID attribute and for Dot1x profile with PVID 4095 is sent from RADIUS with VLAN tunnel ID as 11. Then if both MAC and Dot1x are enabled on the same port, on successful MAC authentication when profile with PVID 0 is applied based on the configurations "Explicit Deny all rule" is installed and Dot1x authentication is succeeded with PVID 4095 (no action) applied, but then traffic does not egress using Tunnel VLAN tag 11 or any default VLAN or admin profile VLAN where the port is bounded since the deny all rule is installed.

Assigning a Class of Service to Policy Role

How a packet is treated as it transits the link can be configured in the *CoS*. It is through a CoS that *QoS (Quality of Service)* is implemented. A CoS can be configured for the following values:

- 802.1p priority
- IP Type of Service (ToS) rewrite value
- Priority Transmit Queue (TxQ) along with a forwarding behavior
- Inbound rate limiter per transmit queue
- Outbound rate shaper per transmit queue

CoS configurations are identified by a numeric value between 0 - 255. 0 - 7 are fixed 802.1p CoS configurations. CoS configurations 8 - 255 are user configurable. Policy uses the cos option followed by the CoS configuration ID value to associate a CoS with a policy role.

Adding Tagged, Untagged, and Forbidden Ports to the VLAN Egress Lists

The *VLAN* egress list contains a list of ports that a frame for this VLAN can exit. Specified ports are automatically assigned to the VLAN egress list for this policy role as tagged, untagged, or forbidden. Ports are added to the VLAN egress list using the

egress-vlans, untagged-vlans, and forbidden-vlans options of the `configure policy profile` command.

Note
Egress policy is not supported.

Overwriting VLAN Tags Priority and Classification Settings

TCI overwrite supports the application of rules to a policy role that overwrite the current user priority and other classification information in the *VLAN* tag's TCI field. TCI overwrite must be enabled for both the policy role and the port the role is applied to.

Use the `configure policy profile` *`profile_index`* `tci-overwrite` command to enable TCI overwrite on a policy role.

Supported Platforms

All ExtremeSwitching Universal platforms.

Classification Rules

Classification rules associate specific traffic classifications or policy behaviors with the policy role. There are two aspects of classification rule configuration:

- The association of a traffic classification with a policy role by assigning the traffic classification to an administrative profile.
- The assignment of policy rules that define desired policy behaviors for the specified traffic classification type.

Both the administrative profile and policy rules are associated with the policy role by specifying the **`admin-pid`** option, in the case of an administrative profile, or a **`profile-index`** value, in the case of the policy rule. Administrative profiles and policy rules are configured using the `configure policy rule` command.

The administrative profile assigns a traffic classification to a policy role by using the **`admin-profile`** option of the `configure policy rule command`.

Note
Standard policy supports the *VLAN* tag traffic classification for administrative profiles. All other traffic classifications are enhanced policy in an administrative profile context.

Note
When both admin-profile port and macsource rule are configured on the same port, macsource rule takes precedence and the VLAN classification and rules applicable to macsource are applied to the port.

Note
When specifying an action without an explicit "forward" or "deny" option, this is interpreted as an implicit permit, since the rule conditions are met. This type of action configuration does not appear in the output of the `show policy` **`access-list`** { [*list_dot_rule*]**`profile-index`** *profile_index*] | [{**`matches`** [**`app-signature`** | **`ether`** | **`icmp6type`** | **`icmptype`** | **`ipdestsocket`** | **`ipfrag`** | **`ipproto`** | **`ipsourcesocket`** | **`iptos`** | **`ipttl`** | **`tcpdestportIP`** | **`tcpsourceportIP`** | **`udpdestportIP`** | **`udpsourceportIP`**] {**`mask`** *mask*} {**`data`** *data*} } {**`actions`** [{**`drop`** | **`forward`** } {**`cos`** *cos*} {**`-1`**} {**`mirror-destination`** *control_index*} {**`syslog`**] }] } {**`detail`**} command.

When admin-profile macsource rule is configured and the port is not added to a VLAN, then the traffic arriving on this port is mapped to the admin-pid configured profile based on the port string. To map it to macsource, add the port to a VLAN.

Policy rules are based on traffic classifications. Table 91 provides the supported policy rule traffic classification command options and definitions. All other traffic classifications are supported by standard policy.

Table 91: Administrative Policy and Policy Rule Traffic Classifications

Traffic Classification	Description	Attribute ID	Enhanced Rule
macsource	Classifies based on MAC source address.	1	
macdest	Classifies based on MAC destination address.	2	
ip6dest	Classifies based on destination IPv6 address.	10	
ipsourcesocket	Classifies based on source IP address.	12	
ipdestsocket	Classifies based on destination IP address.	13	
ip frag	Classifies based on IP fragmentation value.	14	
udpsourceportip	Classifies based on UDP source port and optional post-fix IP address.	15	
udpdestportip	Classifies based on UDP destination port and optional post-fix IP address.	16	
tcpsourceportip	Classifies based on TCP source port and optional post-fix IP address.	17	
tcpdestportip	Classifies based on TCP destination port and optional post-fix IP address.	18	
icmp	Classifies based on ICMP type code.	19	
ipttl	Classifies based on TTL.	20	

Table 91: Administrative Policy and Policy Rule Traffic Classifications (continued)

Traffic Classification	Description	Attribute ID	Enhanced Rule
iptos	Classifies based on Type of Service field in IP packet.	21	
ipproto	Classifies based on protocol field in IP packet.	22	
icmp6	Classifies based on ICMPv6 type code.	23	
ether	Classifies based on type field in Ethernet II packet.	25	
application	Classifies based on Layer 7 DNS snooping.	29	
port	Classifies based on port-string.	31	
IPSourceL4Range	Classifies based on source IP address with post-fixed port-range.	32	
IPDestL4Range	Classifies based on destination IP address with post-fixed port-range.	33	
UDPSrcPortRange	Classifies based on UDP source port-range with optional post-fix IPv4 address.	34	
UDPDestPortRange	Classifies based on UDP destination port-range with optional post-fix IPv4 address.	35	
TCPSrcPortRange	Classifies based on TCP source port-range with optional post-fix IPv4 address.	36	
TCPDestPortRange	Classifies based on TCP destination port-range with optional post-fix IPv4 address.	37	

A data value is associated with most traffic classifications to identify the specific network element for that classification. For data value and associated mask details, see the “Syntax Description” table in the `configure policy rule` *`profile_index`* [{**`app-signature group`** *`group`* **`name`** *`name`*} | **`ether`** *`ether`* | **`icmp6type`** *`icmp6type`* | **`icmptype`** *`icmptype`* | **`ip6dest`** *`ip6dest`* |**`ipdestsocket`** *`ipdestsocket`* | **`ipfrag`** | **`ipproto`** *`ipproto`* | **`ipsourcesocket`** *`ipsourcesocket`* | **`iptos`** *`iptos`* | **`ipttl`** *`ipttl`* | **`macdest`** *`macdest`* | **`macsource`** *`macsource`* | **`port`** *`port`* | **`tcpdestportIP`** *`tcpdestportIP`* | **`tcpsourceportIP`** *`tcpsourceportIP`* | **`udpdestportIP`** *`udpdestportIP`* | **`udpsourceportIP`** *`udpsourceportIP`*] {**`mask`** *`mask`* } {**`port-string`** [*`port_string`* | **`all`**]} {**`storage-type`** [**`non-volatile`** | **`volatile`**]} {**`drop`** | **`forward`**} {**`syslog`** *`syslog`*} {**`trap`** *`trap`*} {**`cos`** *`cos`* } {**`mirror-destination`** *`control_index`*} {**`clear-mirror`**} command in the Switch Engine v33.6.1 Command References .

Specifying Storage Type

ONEPolicy provides for specifying the storage type for this rule entry. Storage types are volatile and non-volatile. Volatile storage does not persist after a reset of the device. Non-volatile storage does persist after a reset of the device. Use the **storage-type** option to specify the desired storage type for this policy rule entry in an enhanced policy context.

Forward and Drop

Packets for this entry can be either forwarded or dropped for this traffic classification using the **forward** and **drop** policy rule options.

Allowed Traffic Rule-Type on a Port

Use the `show policy allowed-type` command to display a table of the current allowed and disallowed traffic rule-types for the specified port(s).

See Table 91 on page 1344 for a listing of supported allowed traffic classification rule-types.

Rule Precedence

Note
When upgrading to ExtremeXOS 30.5 or later, any previously configured (before 30.5) policy profile precedence will be reset to the default rule precedence.

Static rules (MAC + port) have higher precedence than dynamic rules (Dot1x/Mac/Web *VLAN* authorization rules).

Example

```
# configure policy rule admin-profile macsource 00-00-00-00-00-01 mask 48 port-string 27
admin-pid 2

configure policy profile 2 name "filter" pvid-status "enable" pvid 400 egress-vlans 200
untagged-vlans 400 tci-overwrite "enable"

configure policy maptable response both
configure policy vlanauthorization enable
enable policy
```

In the above configuration, if a dot1x user is authenticated with Tunnel Private Group Id as "3000" and filter id as "filter" via Radius, the static macsource rule takes higher precedence and the client *FDB* learned on VLAN SYS_VLAN_0400 mentioned in the static rule rather than the tunnel ID sent by Radius.

```
# show fdb
Mac                     Vlan      Age  Flags           Port / Virtual Port List
-------------------------------------------------------------------------------
00:00:00:00:00:01 SYS_VLAN_0400(0400) 0010 nd m     v     27
```

Modifying Rule Precedence Order

Starting with ExtremeXOS 30.2, you can modify the default precedence of ONEPolicy profile rules. You can configure the precedence for the rule types within each rule group. The rule groups currently supported are MAC, IPv6, IPv4, and Layer2.

The default precedence order is 1–2, 10, 29, 12, 32, 13, 33, 14–15, 34, 16, 35, 17, 36, 18, 37, 19, 23, 20–22, 25, 31.

MACSource (1), MACDest (2), IPv6Dest (10), Application (29), IPSource (12), IPSourceL4Range (32), IPDest (13), IPDestL4Range (33), IPFrag (14), UDPSrcPort (15), UDPSrcPortRange (34), UDPDestPort (16), UDPDestPortRange (35), TCPSrcPort (17), TCPSrcPortRange (36), TCPDestPort (18), TCPDestPortRange (37), ICMPType (19), ICMP6Type (23), TTL (20), IPTOS (21), IPProto (22), Ether (25), Port (31).

Rule types 1–2 come under MAC group.

Rule type 10 comes under IPv6 group.

Rule types 29, 12, 32, 13, 33, 14–15, 34, 16, 35, 17, 36, 18, 37, 19, 23, 20–22 come under IPv4 group.

Rule types 25 and 31 come under L2 group.

To modify the rule precedence order, use the following command with the precedence option:

```
configure policy profile profile_index {name name} {pvid pvid}
{pvid-status pvid_status} {cos cos} {cos-status cos_status} {egress-
vlans egress_vlan_list}{forbidden-vlans forbidden_vlans} {untagged-vlans
untagged_vlans} {append | clear} {tci-overwrite tci_overwrite}
{precedence [precedence | default]} {auth-override auth_override} {nsi
[nsi | none]} {web-redirect web_redir_index} {access-list [unassigned |
list_name | list_name_placeholder]}
```

Quality of Service in a Policy Rules Context

QoS can be specified directly in a policy role as stated in Assigning a Class of Service to Policy Role on page 1342. A *CoS* can also be applied to a policy rule. The CoS specified at the policy role level is the default and is only used if no rule is triggered. Therefore, if a CoS is applied to both the policy role and a policy rule, the CoS specified in the policy rule takes precedence over the CoS in the policy role for the traffic classification context specified in the policy rule. As stated in the policy role discussion, CoS configuration details are beyond the scope of this document.

Blocking Non-Edge Protocols at the Edge Network Layer

Edge clients should be prevented from acting as servers for a number of IP services. If non-edge IP services accidently or maliciously attach to the edge of the network, they are capable of disrupting network operation. IP services should only be allowed where and when your network design requires. This section identifies ten IP Services

you should consider blocking at the edge unless allowing them is part of your network architecture.

Table 92: Non-Edge Protocols

Protocol	Policy Effect
DHCP Server Protocol	Every network needs DHCP. Automatically mitigate the accidental or malicious connection of a DHCP server to the edge of your network to prevent DoS or data integrity issues, by blocking DHCP on the source port for this device.
DNS Server Protocol	DNS is critical to network operations. Automatically protect your name servers from malicious attack or unauthorized spoofing and redirection, by blocking DNS on the source port for this device.
Routing Topology Protocols	*RIP (Routing Information Protocol)*, *OSPF (Open Shortest Path First)*, and *BGP (Border Gateway Protocol)* topology protocols should only originate from authorized router connection points to ensure reliable network operations.
Router Source MAC and Router Source IP Address	Routers and default gateways should not be moving around your network without approved change processes being authorized. Prevent DoS, spoofing, data integrity and other router security issues by blocking router source MAC and router source IP addresses at the edge.
SMTP/POP Server Protocols	Prevent data theft and worm propagation by blocking SMTP at the edge.
SNMP Protocol	Only approved management stations or management data collection points need to be speaking SNMP. Prevent unauthorized users from using SNMP to view, read, or write management information.
FTP and TFTP Server Protocols	Ensure file transfers and firmware upgrades are only originating from authorized file and configuration management servers.
Web Server Protocol	Stop malicious proxies and application-layer attacks by ensuring only the right Web servers can connect from the right location at the right time, by blocking HTTP on the source port for this device.
Legacy Protocols	If IPX, AppleTalk, DECnet or other protocols should no longer be running on your network, prevent clients from using them. Some organizations even take the approach that unless a protocol is specifically allowed, all others are denied.

Platform Rule Allocation

Table 93: Platform Rule Allocation - 4120, 4220, 5120

Table Profile Name	Table Name	4120	4220	5120
Default	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices) 1,024	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices) 1,024
more-ipv4-less-acl	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices)	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices)
more-ipv4-no-ipv6-less-acl	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices)	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices)
more-ipv4-no-mac-no-ipv6-less-acl	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices)	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices)
more-ipv4-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices)	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices)
more-ipv4-no-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices)	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices)

Table 93: Platform Rule Allocation - 4120, 4220, 5120 (continued)

Table Profile Name	Table Name	4120	4220	5120
more-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 128 56 1,024 (4 slices)	0 0 256 184 1,024 (4 slices) 1,024	0 0 128 56 1,024 (4 slices)
more-ipv4-no-l2-less-acl- *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 184 0 1,024 (4 slices)	0 0 440 0 1,024 (4 slices)	0 0 184 0 1,024 (4 slices)
more-ipv4-no-mac-no-ipv6-no-l2-less-acl- *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 184 0 1,024 (4 slices)	0 0 440 0 1,024 (4 slices)	0 0 184 0 1,024 (4 slices)
more-ipv4-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 184 0 1,024 (4 slices)	0 0 440 0 1,024 (4 slices)	0 0 184 0 1,024 (4 slices)

Table 93: Platform Rule Allocation - 4120, 4220, 5120 (continued)

Table Profile Name	Table Name	4120	4220	5120
more-ipv4-no-mac-no-ipv6-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/ System Rules	0 0 184 0 1,024 (4 slices)	0 0 440 0 1,024 (4 slices)	0 0 184 0 1,024 (4 slices)
* The L2 rule group normally includes ether-type and port qualifiers. When no-l2 is used, all the ether rules are accounted for in the IPv4 group. If you remove both L2 and IPv4 (no-l2 and no-ipv4), then ether rules are accounted for in the IPv6 group. If you remove L2, IPv4, and IPv6, (no-l2, no-ipv4, no-ipv6), then ether rules are accounted for in the MAC group. **Note:** The best possible ExtremeXOS system application/system rules are driven by removing the default system ACL rules (`disable dot1p replacement ports all`), while keeping these system default ACL rules. **Note:** Free ACL-slices are required (excluding the policy utilized slices) for MLAG to function properly. **`configure policy resource-profile`** is required.				

Table 94: Platform Rule Allocation - 5320, 5420, 5520

Table Profile Name	Table Name	5320-48T	5320-16P	5320-24T-4X-XT	5320-24T-24S-4XE-XT	5420M	5420F	5520
Default	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	512 512 512 440 1,024 (4 slices)	0 0 1,024 952 1,024 (4 slices)	0 0 256 184 1,024 (4 slices)	512 512 512 440 1,024 (4 slices)	1,024 1,024 1,024 952 1,024 (4 slices)	512 512 512 440 1,024 (4 slices)	1,024 1,024 1,024 952 4,096 (4 slices) / 1,024 rules
more-ipv4-less-acl	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	512 512 512 440	0 0 1,024 952	0 0 256 184 1,024 (4 slices)	512 512 512 440 1,024 (4 slices)	1,024 1,024 1,024 952 1,024 (4 slices)	512 512 512 440 1,024 (4 slices)	1,024 1,024 1,024 952 4,096 (4 slices)

Table 94: Platform Rule Allocation - 5320, 5420, 5520 (continued)

Table Profile Name	Table Name	5320-48T	5320-16P	5320-24T-4X-XT	5320-24T-24S-4XE-XT	5420M	5420F	5520
								/ 1,024 rules
more-ipv4-no-ipv6-less-acl	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	512 0 1,024 440	0 0 1,024 952	0 0 256 184 1,024 (4 slices)	512 0 1024 440 1,024 (4 slices)	1,024 0 2048 952 1,024 (4 slices)	512 0 1,024 440 1,024 (4 slices)	1,024 0 2,048 952 4,096 (4 slices) / 1,024 rules
more-ipv4-no-mac-no-ipv6-less-acl	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	0 0 1536 440	0 0 1,024 952	0 0 256 184 1,024 (4 slices)	0 0 1536 440 1,024 (4 slices)	0 0 3072 952 1,024 (4 slices)	0 0 1536 440 1,024 (4 slices)	0 0 3,072 952 4,096 (4 slices) / 1,024 rules
more-ipv4-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	512 0 1,024 440	0 0 1,024 952	0 0 256 184 1,024 (4 slices)	512 0 1024 440 1,024 (4 slices)	1,024 0 2048 952 1,024 (4 slices)	512 0 1,024 440 1,024 (4 slices)	1,024 0 2,048 952 4,096 (4 slices) / 1,024 rules
more-ipv4-no-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	0 0 1536 440	0 0 1,024 952	0 0 256 184 1,024 (4 slices)	0 0 1536 440 1,024 (4 slices)	0 0 3072 952 1,024 (4 slices)	0 0 1536 440 1,024 (4 slices)	0 0 3,072 952 4,096 (4 slices) / 1,024 rules

Table 94: Platform Rule Allocation - 5320, 5420, 5520 (continued)

Table Profile Name	Table Name	5320-48T	5320-16P	5320-24T-4X-XT	5320-24T-24S-4XE-XT	5420M	5420F	5520
more-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	1,024 0 512 440	0 0 1,024 952	0 0 256 184 1,024 (4 slices)	1024 0 512 440 1,024 (4 slices)	2048 0 1,024 952 1,024 (4 slices)	1,024 0 512 440 1,024 (4 slices)	2,048 0 1,024 952 4,096 (4 slices) / 1,024 rules
more-ipv4-no-l2-less-acl- *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	N/A	N/A	0 0 440 0 1,024 (4 slices)	512 512 952 0 1,024 (4 slices)	1,024 1,024 1976 0 1,024 (4 slices)	512 512 952 0 1,024 (4 slices)	1,024 1,024 1,976 0 4,096 (4 slices) / 1,024 rules
more-ipv4-no-mac-no-ipv6-no-l2-less-acl- *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	N/A	N/A	0 0 440 0 1,024 (4 slices)	0 0 1976 0 1,024 (4 slices)	0 0 4024 0 1,024 (4 slices)	0 0 1976 0 1,024 (4 slices)	0 0 4,024 0 4,096 (4 slices) / 1,024 rules
more-ipv4-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules			0 0 440 0 1,024 (4 slices)	512 512 952 0 1,024 (4 slices)	1,024 1,024 1976 0 1,024 (4 slices)	512 512 952 0 1,024 (4 slices)	1,024 1,024 1,976 0 4,096 (4 slices) / 1,024 rules

Table 94: Platform Rule Allocation - 5320, 5420, 5520 (continued)

Table Profile Name	Table Name	5320-48T	5320-16P	5320-24T-4X-XT	5320-24T-24S-4XE-XT	5420M	5420F	5520
more-ipv4-no-mac-no-ipv6-no-l2 *	MAC Rules			0	0	0	0	0
	IPv6 Rules			0	0	0	0	0
	IPv4 Rules			440	1976	4024	1976	4,024
	L2 Rules			0	0	0	0	0
	ExtremeXOS App/System Rules			1,024 (4 slices)	1,024 (4 slices)	1,024 (4 slices)	1,024 (4 slices)	4,096 (4 slices) / 1,024 rules
* The L2 rule group normally includes ether-type and port qualifiers. When no-l2 is used, all the ether rules are accounted for in the IPv4 group. If you remove both L2 and IPv4 (no-l2 and no-ipv4), then ether rules are accounted for in the IPv6 group. If you remove L2, IPv4, and IPv6, (no-l2, no-ipv4, no-ipv6), then ether rules are accounted for in the MAC group. **Note:** The best possible ExtremeXOS system application/system rules are driven by removing the default system ACL rules (`disable dot1p replacement ports all`), while keeping these system default ACL rules. **Note:** Free ACL-slices are required (excluding the policy utilized slices) for MLAG to function properly. **`configure policy resource-profile`** is required.								

Table 95: Platform Rule Allocation - 5720, 7520, 7720

Table Profile Name	Table Name	5720-MXW	5720-MW	7520-48Y-8C	7520-48XT-6C	7720-32-C
Default	MAC Rules	2,048	1,536	512	512	512
	IPv6 Rules	2,048	1,536	512	512	512
	IPv4 Rules	2,048	1,536	512	512	512
	L2 Rules	1,976	1,464	440	440	440
	ExtremeXOS App/System Rules	20,480 (10 slices) / 2,048 rules	6,144 (4 slices) / 1,536 rules	2,048 (4 slices)	2,048 (4 slices)	2,048 (4 slices)
less-acl-more-ipv4	MAC Rules	2,048	1,536	512	512	512
	IPv6 Rules	2,048	1,536	512	512	512
	IPv4 Rules	2,048	1,536	512	512	512
	L2 Rules	1,976	1,464	440	440	440
	ExtremeXOS App/System Rules	20,480 (10 slices) / 2,048 rules	6,144 (4 slices) / 1,536 rules			

Table 95: Platform Rule Allocation - 5720, 7520, 7720 (continued)

Table Profile Name	Table Name	5720-MXW	5720-MW	7520-48Y-8C	7520-48XT-6C	7720-32-C
less-acl-more-ipv4-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	2,048 0 4,096 1,976 20,480 (10 slices) / 2,048 rules	1,536 0 3,072 1,464 6,144 (4 slices) / 1,536 rules	512 0 1,024 440	512 0 1,024 440	512 0 1,024 440
less-acl-more-ipv4-no-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	0 0 6,144 1,976 20,480 (10 slices) / 2,048 rules	0 0 4,608 1,464 6,144 (4 slices) / 1,536 rules	0 0 1,536 440	0 0 1,536 440	0 0 1,536 440
more-ipv4-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	2,048 0 4,096 1,976 20,480 (10 slices) / 2,048 rules	1,536 0 3,072 1,464 6,144 (4 slices) / 1,536 rules	510 0 1,024 440	510 0 1,024 440	510 0 1,024 440
more-ipv4-no-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	0 0 6,144 1,976 20,480 (10 slices) / 2,048 rules	0 0 4,608 1,464 6,144 (4 slices) / 1,536 rules	0 0 1,536 440 2,048 (4 slices)	0 0 1,536 440 2,048 (4 slices)	0 0 1,536 440 2,048 (4 slices)
more-mac-no-ipv6	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	4,096 0 2,048 1,976 20,480 (10 slices) / 2,048 rules	3,072 0 1,536 1,464 6,144 (4 slices) / 1,536 rules	1,024 0 512 440 2,048 (4 slices)	1,024 0 512 440 2,048 (4 slices)	1,024 0 512 440 2,048 (4 slices)
less-acl-more-ipv4-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	2,048 2,048 4,096 0 20,480 (10 slices) / 2,048 rules	1,536 1,536 3,000 0 6,144 (4 slices) / 1,536 rules	512 512 952 0	512 512 952 0	512 512 952 0

Table 95: Platform Rule Allocation - 5720, 7520, 7720 (continued)

Table Profile Name	Table Name	5720-MXW	5720-MW	7520-48Y-8C	7520-48XT-6C	7720-32-C
less-acl-more-ipv4-no-mac-no-ipv6-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	0 0 8,192 0 20,480 (10 slices) / 2,048 rules	0 0 6,072 0 6,144 (4 slices) / 1,536 rules	0 0 1,976 0	0 0 1,976 0	0 0 1,976 0
more-ipv4-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	2,048 2,048 4,096 0 20,480 (10 slices) / 2,048 rules	1,536 1,536 3,000 0 6,144 (4 slices) / 1,536 rules	512 512 952 0	512 512 952 0	512 512 952 0
more-ipv4-no-mac-no-ipv6-no-l2 *	MAC Rules IPv6 Rules IPv4 Rules L2 Rules ExtremeXOS App/System Rules	0 0 8,192 0 20,480 (10 slices) / 2,048 rules	0 0 6,072 0 6,144 (4 slices) / 1,536 rules	0 0 1,976 0 2,048 (4 slices)	0 0 1,976 0 2,048 (4 slices)	0 0 1,976 0 2,048 (4 slices)
* The L2 rule group normally includes ether-type and port qualifiers. When no-l2 is used, all the ether rules are accounted for in the IPv4 group. If you remove both L2 and IPv4 (no-l2 and no-ipv4), then ether rules are accounted for in the IPv6 group. If you remove L2, IPv4, and IPv6, (no-l2, no-ipv4, no-ipv6), then ether rules are accounted for in the MAC group. **Note:** The best possible ExtremeXOS system application/system rules are driven by removing the default system ACL rules (`disable dot1p replacement ports all`), while keeping these system default ACL rules. **Note:** Free ACL-slices are required (excluding the policy utilized slices) for MLAG to function properly. **`configure policy resource-profile`** is required.						

For information about the maximum authenticated users per switch, see the Limits table of the *ExtremeXOS Release Notes*.

In ExtremeXOS 22.4, you can use the profile modifier feature to return resources back to ACL from the specified profile. Use the profile-modifier option in the following command:

```
configure policy resource-profile [default |less-acl [more-ipv4 | more-ipv4-no-ipv6 | more-ipv4-no-l2 |more-ipv4-no-mac-no-ipv6] | more-ipv4-no-mac-no-ipv6-no-l2 | more-ipv4-no-ipv6 | more-ipv4-no-mac-no-ipv6 | more-mac-no-ipv6] {profile-modifier [{no-mac no_mac} {no-ipv4 no_ipv4} {no-ipv6 no_ipv6} {no-l2 no_l2}]}
```

To see what profile modifier settings you have configured, use the following command:

```
show policy resource-profile {[default | less-acl [more-ipv4 |more-ipv4-no-ipv6 | more-ipv4-no-l2 |more-ipv4-no-mac-no-ipv6 | more-ipv4-no-mac-no-ipv6-no-l2] |more-ipv4-no-ipv6 | more-ipv4-no-mac-no-ipv6 | more-mac-no-ipv6] {profile-modifier [ {no-mac} {no-ipv4} {no-ipv6} {no-l2}]}}
```

The following tables show the maximum number of slices available with the resource profile modifier enabled.

Table 96: Slices Available with Resource Profile Modifier for ExtremeSwitching Universal Switches

Profile Name	Max Slices Available	No Profile Modifier	No-mac	No-ipv4	No-ipv6	No-l2	No-mac, no-ipv4	No-mac, no-ipv6	no-mac-no-l2	No-ipv4, no-ipv6	no-ipv4-no-l2	no-ipv6-no-l2	no-mac-no-ipv4-no-ipv6	no-mac-no-ipv4-no-l2	no-mac-no-ipv6-no-l2	no-ipv4-no-ipv6-no-l2
Default	4	4	3	3	3	3	2	2	2	2	2	2	1	1	1	1
less-acl more-ipv4	4	4	3	3	3	3	2	2	2	2	2	2	1	1	1	1
less-acl-more-ipv4-no-ipv6	4	4	3	2	4	3	1	3	2	2	1	3	1	N/A	2	1
less-acl-more-ipv4- no-l2	4	4	3	2	3	4	1	2	3	1	2	3	N/A	1	2	1
less-acl-more-ipv4-no-mac-no-ipv6	4	4	4	1	4	3	1	4	3	1	N/A	3	1	N/A	3	N/A

Table 96: Slices Available with Resource Profile Modifier for ExtremeSwitching Universal Switches (continued)

Profile Name	Max Slices Available	No Profile Modifier	No-mac	No-ipv4	No-ipv6	No-l2	No-mac, no-ipv4	No-mac, no-ipv6	no-mac- no-l2	No-ipv4, no-ipv6	no-ipv4-no-l2	no-ipv6-no-l2	no-mac-no-ipv4-no-ipv6	no-mac-no-ipv4-no-l2	no-mac-no-ipv6-no-l2	no-ipv4-no-ipv6-no-l2
less-acl-more-ipv4-no-mac-no-ipv6-no-l2	4	4	4	N/A	4	4	N/A	4	4	N/A	N/A	4	N/A	N/A	4	N/A
more-ipv4-no-ipv6	4	4	3	2	4	3	1	3	2	2	1	3	1	N/A	2	1
more-ipv4-no-l2	4	4	3	2	3	4	1	2	3	1	2	3	N/A	1	2	1
more-ipv4-no-mac-no-ipv6	4	4	4	1	4	3	1	4	3	1	N/A	3	1	N/A	3	N/A
more-ipv4-no-mac-no-ipv6-no-l2	4	4	4	N/A	4	4	N/A	4	4	N/A	N/A	4	N/A	N/A	4	N/A
more-mac-no-ipv6	4	4	2	3	4	3	1	2	1	3	2	3	1	N/A	1	2

Authentication and ONEPolicy

Applying Policy Using the RADIUS Response Attributes

If an authentication method that requires communication with an authentication server is configured for a user, the *RADIUS* filter-ID attribute can be used to dynamically assign a policy role to the authenticating user. Supported RADIUS attributes are sent to the switch in the RADIUS access-accept message. The RADIUS filter-ID can also be applied in hybrid authentication mode. Hybrid authentication mode determines how the RADIUS filter-ID and the three RFC 3580 *VLAN* tunnel attributes (VLAN Authorization), when either or all are included in the RADIUS access-accept message, will be handled by the switch. The three VLAN tunnel attributes define the base VLAN-ID to be applied to the user. In either case, conflict resolution between RADIUS attributes is provided by the maptable response feature.

Note
The maptable response feature is only applicable if VLAN Authorization is enabled (`configure policy vlanauthorization enable`).

Note
VLAN-to-policy mapping to maptable response configuration behavior is as follows:

- If the RADIUS response is set to policy, any VLAN-to-policy maptable configuration is ignored for all platforms.
- If the RADIUS response is set to both and both the filter-ID and tunnel attributes are present, VLAN-to-policy mapping configuration is ignored. See the “When Policy Maptable Response is Both” section of the Configuring User Authentication feature guide for exceptions to this behavior.

Use the policy option of the `configure policy maptable response` command to configure the switch to dynamically assign a policy using the RADIUS filter-ID in the RADIUS response message.

Table 97 shows the RADIUS access-accept attributes for ONEPolicy that ExtremeXOS supports.

Table 97: Supported Access-Accept Attributes for ONEPolicy

Access-Accept Attribute	Description	Notes
Filter-Id	Policy Profile Name	
Tunnel-Medium-Type	IEEE-802	Must be present when using Tunnel-Private-Group-Id.
Tunnel-Type	VLAN	Must be present when using Tunnel-Private-Group-Id.

Table 97: Supported Access-Accept Attributes for ONEPolicy (continued)

Access-Accept Attribute	Description	Notes
Tunnel-Private-Group-Id	Tunnel-ID	Can be a VLAN tag or the pre-configured tagged VLAN name (string).
Session-Timeout	Numbers in seconds	'0' if it is not present.
Idle-Timeout	Numbers in seconds	'0' if it is not present.
Termination-action	Default/Radius | 0/1	Default if it is not present.
Fabric-Attach-VLAN-ISID	Tunnel-ID:4digitNSID+4digitTunnel-ID	Example: 10:12010010.

Multi-Host Single Authentication

Version 33.1 adds the Multi-Host Single Authentication (MHSA) feature that allows a single authenticated device to authorize all other devices connected through it, without requiring each one to authenticate individually.

When an AP authenticates to a switch port using RADIUS and the MHSA VSA is included in the RADIUS response, the switch automatically enables **authentication override** on the dynamically created policy profile. This means any additional hosts connected through the AP can access the network without separate authentication. Once the last authenticated user disconnects, the override is removed and the profile is deleted.

This behavior mirrors functionality previously available in Switch Engine, where a policy profile could be configured to allow multi-host access after the first successful authentication.

VSA Format

Only an integer value of `1` is supported.

Example

In this example:

- If the profile `myprofile` does not already exist on the switch, it will be **dynamically created** using the downloadable ACL flags (`t:a,rc`).
- Because the **MHSA VSA** is present (`Extreme-Dyn-MHSA = "1"`), the profile will be created with **authentication override enabled**.

```
Extreme-Policy-ACL += "v:1 t:a,rc
m:ipv4dst=28.0.0.1/32,ipproto=udp,l4srcport=102-103,l4dstport=201-250 a:fwd,sys ",
Extreme-Policy-ACL += "v:1 t:a,rc m:ipproto=icmp a:fwd,sys ",
Extreme-Policy-ACL += "v:1 t:a,rc m:any a:drop,sys ",
Extreme-Dyn-MHSA = "1",
Filter-Id = "Enterasys:version=1:mgmt=rw:policy=myprofile"
```

Important behavior notes

- The **authentication override setting is locked** at the time the profile is created.
- If the first user authenticates **without** the MHSA VSA, then **subsequent users cannot enable it**, even if their entries include the VSA.
- To avoid conflicts, all users authenticating to the same profile should **consistently include or exclude** the MHSA VSA.

NetLogin Authentication

NetLogin provides the dynamic authentication of users as a frontend to policy. Supported authentication type for ExtremeXOS includes 8021.1X, MAC Authentication, and Web Authentication.

Unknown unicast/multicast/broadcast traffic is allowed to egress NetLogin-enabled ports, even if the ports are not authenticated. Configuring the port authentication mode as optional/required does not affect egress traffic (`configure netlogin ports [all | port_list] [allowed-users allowed_users | authentication mode [optional | required] | trap [all-traps | no-traps | [{success} {failed} {terminated} {max-reached}]]]`).

Note
When enabling policy, all *VLAN*-level commands supported in non-policy mode are lost, including:

- `configure netlogin vlan`
- `configure netlogin authentication failure vlan`
- `configure netlogin authentication service-unavailable vlan`
- `configure netlogin dot1x guest-vlan`
- `configure netlogin dynamic-vlan`
- `configure netlogin ports [all | port_list] mode [mac-based-vlans | port-based-vlans]`
- `configure netlogin ports [all | port_list] no-restart`
- `configure netlogin ports [all | port_list] restart`
- `configure netlogin ports [port_list | all] allow egress-traffic [none | unicast | broadcast | all_cast]`

NetLogin Session Timeout and Idle Timeout

Use the following commands to configure session timeout and idle timeout locally. These commands take effect if RADIUS access-accept has not returned any session timeout/idle timeout:

`configure netlogin session-timeout {`**`dot1x`** | **`mac`** | **`web-based`** | **`convergence-endpoint`**`} timeout`

Note
If you want to scale to 65,000 authenticated users, use a session timeout value of at least 300 minutes.

`configure netlogin idle-timeout {`**`convergence-endpoint`** | **`dot1x`** | **`mac`** | **`web-based`**`} timeout`

These commands appear in `show configuration {module-name} {`**`detail`**`}` for "policy" rather than "netlogin," since they are specific to ONEPolicy mode.

The command `show netlogin session {`**`all`** | **`summary`**`} {`**`mac-address`** *mac_address*`} {`**`ports`** *ports*`} {`**`agent`** `[`**`convergence-endpointdot1x`** | **`mac`** | **`web-based`**`]}` shows the session timeout and idle timeout, and `show netlogin mac` does not show any reauth/session timeout values.

```
# show netlogin session
 Multiple authentication session entries
 ---------------------------------------

 Port            : 1:1         Station address   : 00:00:03:00:00:00
 Auth status     : success     Last attempt      : Tue May 23 08:24:17 2017
 Agent type      : mac         Session applied   : true
 Server type     : radius      VLAN-Tunnel-Attr  : None
 Policy index    : 1           Policy name       : Extreme (active)
 Session timeout : 40          Session duration  : 0:00:02
 Idle timeout    : 20          Idle time         : 0:00:00
 Auth-Override   : enabled     Termination time: Not Terminated

# show netlogin port 1:1
 Port                          : 1:1
 Authentication                : mac-based
 Port State                    : Enabled
 Authentication Mode           : Required (Policy Enabled only)
 Max Supported Users           : 1024 (Policy Enabled only)
 Allowed Users                 : 1024 (Policy Enabled only)
 Current Users                 : 2 (Policy Enabled only)
 ------------------------------------------------
         MAC Mode Port Configuration
 ------------------------------------------------
 Re-authentication period      : 3600
 Re-authentication             : Off
 Authentication Delay          : 0 seconds (Default)
 ------------------------------------------------
         Netlogin Clients
 ------------------------------------------------

 MAC               IP address       Authenticated     Type    ReAuth-Timer   User
 00:00:03:00:00:00  0.0.0.0         Yes, Radius       MAC     0              000003000000
 00:00:03:00:00:01  0.0.0.0         Yes, Radius       MAC     0              000003000001
 ------------------------------------------------
 (B) - Client entry Blackholed in FDB


 Number of Clients Authenticated  : 2
```

When idle timeout is configured and if the FDB is removed, the show netlogin session and `show netlogin port / mac/dot1x/web-based` commands show the NetLogin

authenticated entries untill the idle timer expires. NetLogin session and NetLogin MAC/dot1x/web table is cleared only after the idle timer expires.

Applying Policy Using Hybrid Authentication Mode

Hybrid authentication is an authentication capability that allows the switch to use both the filter-ID and tunnel attributes in the *RADIUS* response message to determine how to treat the authenticating user. Hybrid authentication is configured by specifying the **both** option in the `configure policy maptable response` command. The **both** option:

- Applies the *VLAN* tunnel attributes if they exist and the filter-ID attribute does not exist
- Applies the filter-ID attribute if it exists and the VLAN tunnel attributes do not exist
- Applies both the filter-ID and the VLAN tunnel attributes if all attributes exist

  If all attributes exist, the following rules apply:
  - The policy role will be enforced, with the exception that any port PVID specified in the role will be replaced with the VLAN tunnel attributes
  - The policy map is ignored because the policy role is explicitly assigned
  - VLAN classification rules are assigned as defined by the policy role

vlanauthorization must be enabled or the VLAN tunnel attributes are ignored and the default VLAN is used.

When policy maptable response is set to **both** and only Tunnel ID is returned from RADIUS server, tunnel ID takes precedence and FDB is learned on Tunnel ID if policy maptable is not configured on the switch. If policy maptable is configured, then the policy profile assigned to that VLAN ID takes precedence and FDB is learned on policy profile PVID and not VLAN tunnel ID if invalid action is set to default-policy/drop.

For example:

```
configure policy profile 60 name test pvid 2 pvid-status enable
configure policy maptable 1234 60
```

From RADIUS VLAN tunnel ID 1234 exclusively is sent. Now FDB after successful authentication is learned on PVID 2 and not on 1234.

Hybrid Mode support eliminates the dependency of VLAN assignment based on roles. As a result, VLANs can be assigned via the tunnel-private-group-ID, as defined per RFC3580, while assigning roles via the filter-ID. This separation gives administrators more flexibility to segment their networks for efficiency beyond the role limits.

Authentication Override

Authentication override allows you to override port authentication using a profile-based attribute. If a port has an active policy and the authentication override is enabled, all frames arriving on that port have that policy applied, and no further authentication occurs. In addition, any pre-existing authenticated sessions on that

port are removed. However, the action is reverted once the authentication override is disabled. Authentication override is disabled by default.

The ENTERASYS-POLICY-PROFILE-MIB is changed to support the policy profile-based port authentication override feature:

```
etsysPolicyProfilePortAuthOverride OBJECT-TYPE
      SYNTAX      EnabledStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
            "If a port has an active policy and that policy's
etsysPolicyProfilePortAuthOverride is set to enabled(1),
            all frames arriving on the port will have that policy applied. In addition,
any pre-existing entries with matching
            port values in the etsysMultiAuthSessionStationTable tables will change their
authorization status to
            authTerminated(5).  No further authentication will occur on this port.
            If disabled(2), the actions described above will not occur."
      DEFVAL { disabled }
      ::= { etsysPolicyProfileEntry 21 }

etsysPolicyClassification group
              supportsProfilePortAuthOverride(24)
               -- supports per profile port authentication
            -- override via etsysPolicyProfilePortAuthOverride
```

To configure authentication override, use the following command:

`configure policy profile` *profile_index* {**name** *name*} {**pvid** *pvid*} {**pvid-status** *pvid_status*} {**cos** *cos*} {**cos-status** *cos_status*} {**egress-vlans** *egress_vlan_list*}{**forbidden-vlans** *forbidden_vlans*} {**untagged-vlans** *untagged_vlans*} {**append** | **clear**} {**tci-overwrite** *tci_overwrite*} {**precedence** [*precedence* | **default**]} {**auth-override** *auth_override*} {**nsi** [*nsi* | **none**]} {**web-redirect** *web_redir_index*} {**access-list** [**unassigned** | *list_name* | *list_name_placeholder*]}

Note
If authentication override is enabled, then static *VLAN* has to be configured rather than using the dynamic VLAN for the PVID.

Configuring Policy

This section presents configuration procedures and tables including command description and syntax in the following policy areas: profile, classification, and display.

Table 98: Configuring Policy Roles and Related Functionality

Step	Task	Commands
1	Create a policy role. • name – (Optional) Specifies a name for this policy profile; used by the filter-ID attribute. This is a string from 1 to 64 characters. • pvid-status – (Optional) Enables or disables PVID override for this policy profile. If all the classification rules associated with this profile are missed, then this parameter, if specified, determines the default *VLAN* for this profile. • pvid – (Optional) Specifies the PVID to assign to packets, if PVID override is enabled and invoked as the default behavior. • cos-status – (Optional) Enables or disables Class of Service override for this policy profile. If all the classification rules associated with this profile are missed, then this parameter, if specified, determines the default *CoS* assignment. • cos – (Optional) Specifies a CoS value to assign to packets, if CoS override is enabled and invoked as the default behavior. Valid values are 0 to 255. • egress-vlans – (Optional) Specifies the port to which this policy profile is applied should be added to the egress list of the VLANs defined by egress-vlans. Packets will be formatted as tagged. **Note:** Egress policy is not supported in ExtremeXOS 16.1. • forbidden-vlans - (Optional) Specifies the port to which this policy profile is applied should be added as forbidden to the egress list of the VLANs defined by forbidden-vlans. Packets from this port are not allowed to participate in the listed VLANs.	`configure policy profile` *profile_index* {**name** *name*} {**pvid** *pvid*} {**pvid-status** *pvid_status*} {**cos** *cos*} {**cos-status** *cos_status*} {**egress-vlans** *egress_vlan_list*} {**forbidden-vlans** *forbidden_vlans*} {**untagged-vlans** *untagged_vlans*} {**append** | **clear**} {**tci-overwrite** *tci_overwrite*} {**precedence** [*precedence* | **default**]} {**auth-override** *auth_override*} {**nsi** [*nsi* | **none**]} {**web-redirect** *web_redir_index*} {**access-list** [**unassigned** | *list_name* | *list_name_placeholder*]}

Table 98: Configuring Policy Roles and Related Functionality (continued)

Step	Task	Commands
	• untagged-vlans – (Optional) Specifies the port to which this policy profile is applied should be added to the egress list of the VLANs defined by untagged-vlans. Packets will be formatted as untagged. • append – (Optional) Appends any egress, forbidden, or untagged specified VLANs to the existing list. If append is not specified, all previous settings for this VLAN list are replaced • clear – (Optional) Clears any egress, forbidden or untagged VLANs specified from the existing list. • tci-overwrite – (Optional) Enhanced policy that enables or disables TCI (Tag Control Information) overwrite for this profile. When enabled, rules configured for this profile are allowed to overwrite user priority and other classification information in the VLAN tag's TCI field. • precedence - (Optional) Modify the default precedence of ONEPolicy profile rules. • auth-override - (Optional) If a port has an active policy and the authentication override is enabled, all frames arriving on that port have that policy applied, and no further authentication occurs.	
2	Optionally, for enhanced policy capable devices, assign the action the device will apply to an invalid or unknown policy. • default-policy – Instructs the device to ignore this result and search for the next policy assignment rule. • drop – Instructs the device to block traffic. • forward – Instructs the device to forward traffic.	`configure policy invalid action {`**`default-policy`** `|` **`drop`** `|` **`forward`**`}`

Table 98: Configuring Policy Roles and Related Functionality (continued)

Step	Task	Commands
3	Optionally, for enhanced policy capable devices, set a policy maptable entry that associates a VLAN with a policy profile.	`configure policy maptable {vlan-list profile-index}`
4	Optionally, set a policy maptable response. • tunnel – Applies the VLAN tunnel attribute. • policy – Applies the policy specified in the filter-ID. • both – An enhanced policy option that applies either or all the filter-ID and VLAN tunnel attributes or the policy depending upon whether one or both are present.	`configure policy maptable response {tunnel | policy | both}`

Table 99: Configuring Classification Rules as an Administrative Profile or to Assign Policy Rules to a Policy Role

Step	Task	Command(s)
1	Optionally set an administrative profile to assign traffic classifications to a policy role. See Table 91 on page 1344 for traffic classification-type descriptions and enhanced policy information. See the set policy rule command discussion in the command reference guide that comes with your device for traffic classification data and mask information. • port-string – Applies this administratively-assigned rule to a specific ingress port. The `configure policy port` command is also supported as an alternative way to administratively assign a profile rule to a port. • storage-type – (Optional) Adds or removes this entry from non-volatile storage. • admin-pid – Associates this administrative profile with a policy profile index ID. Valid values are 1 - 1023.	`configure policy rule admin-profile [ macsource macsource | port port ] {mask mask } {port-string [port_string | all] } {storage-type [non-volatile | volatile]} {admin-pid admin_pid }`
2	Optionally configure policy rules to associate with a policy role. See Table 91 on page 1344 for traffic classification-type descriptions and enhanced policy information. See the	`configure policy rule profile_index [{app-signature group group name name} | ether ether | icmp6type icmp6type | icmptype icmptype |`

Table 99: Configuring Classification Rules as an Administrative Profile or to Assign Policy Rules to a Policy Role (continued)

Step	Task	Command(s)
	`configure policy rule` command discussion in the command reference guide that comes with your device for traffic classification data and mask information. • port-string – (Optional) Applies this policy rule to a specific ingress port. The `set policy port` command is also supported as an alternative way to assign a profile rule to a port. • storage-type – (Optional) Adds or removes this entry from non-volatile storage. • drop | forward – (Optional) Specifies that packets within this classification will be dropped or forwarded. • cos – (Optional) Specifies that this rule will classify to a Class-of-Service ID. Valid values are 0 - 255. A value of -1 indicates that no CoS forwarding behavior modification is desired.	**ip6dest** *ip6dest* |**ipdestsocket** *ipdestsocket* | **ipfrag** | **ipproto** *ipproto* | **ipsourcesocket** *ipsourcesocket* | **iptos** *iptos* | **ipttl** *ipttl* | **macdest** *macdest* | **macsource** *macsource* | **port** *port* | **tcpdestportIP** *tcpdestportIP* | **tcpsourceportIP** *tcpsourceportIP* | **udpdestportIP** *udpdestportIP* | **udpsourceportIP** *udpsourceportIP*] {**mask** *mask* } {**port-string** [*port_string* | **all**]} {**storage-type** [**non-volatile** | **volatile**]} {**drop** | **forward**} {**syslog** *syslog*} {**trap** *trap*} {**cos** *cos* } {**mirror-destination** *control_index*} {**clear-mirror**}
3	Optionally, for enhanced policy capable devices, assign a policy role to a port.	`configure policy port` *ports* `admin-id` *admin_id*

Table 100: Displaying Policy Information and Statistics

Step	Task	Command(s)
1	Display policy role information.	`show policy profile` {**all** | profile-index [**-detail**]}
2	Display the action the device should take if asked to apply an invalid or unknown policy, or the number of times the device has detected an invalid/unknown policy, or both action and count information.	`show policy invalid` {**action** | **count** | **all**}
3	Display VLAN-ID to policy role mappings table.	`show policy maptable` [*vlan-list*]

Table 100: Displaying Policy Information and Statistics (continued)

Step	Task	Command(s)
4	Display policy classification and admin rule information.	`show policy rule {all | app-signature | {profile-index profile_index | admin-profile} ether {ether} | icmp6type {icmp6type} | icmptype {icmptype} | ip6dest {ip6dest} | ipdest {ipdest} | ipfrag | ipproto {ipproto} | ipsource { ipsource } | iptos { iptos } | ipttl { ipttl } | macdest { macdest } | macsource { macsource } | port { port } | tcpdestportIP { tcpdestportIP } | tcpsourceportIP { tcpsourceportIP } | udpdestportIP { udpdestportIP } | udpsourceportIP { udpsourceportIP }} {mask mask } {port-string [port_string | all]} {storage-type [non-volatile | volatile]} {drop | forward} {cos cos | admin-pid admin_pid }} {detail | wide}`
5	Display all policy classification capabilities for this device.	`show policy capability`
6	Display a list of currently supported traffic rules applied to the administrative profile for one or more ports.	`show policy allowed-type ports [detail]`
7	Display status of dynamically assigned roles and the current status that the default of dynamically created rules will have in sending of Syslog messages or traps on rule applied.	`show policy dynamic [override | syslog-default | trap-default ]`
8	Display the Syslog parameters for policy rules.	`show policy syslog {machine-readable} {extended-format} {every-time}`
9	Display the interval at which the switch automatically clears rule usage statistics.	`show policy autoclear interval`

Table 100: Displaying Policy Information and Statistics (continued)

Step	Task	Command(s)
10	Display rule usage information when Syslog or trap actions have been set.	`show policy rule port-hit {data} {detail} {wide}`
11	Display captive portal settings.	`show policy captive-portal {web-redirect {redirect_index | all} | listening | rule-use}`
12	Display policy application signature information.	`show policy app-signature group {group {name name}} {built-in | custom {detail} | detail}`
13	Display the existing usage of policy slices.	`show policy slices`
14	Display access list information.	`show policy access-list { [list_dot_ruleprofile-index profile_index] | [{matches [app-signature | ether | icmp6type | icmptype | ipdestsocket | ipfrag | ipproto | ipsourcesocket | iptos | ipttl | tcpdestportIP | tcpsourceportIP | udpdestportIP | udpsourceportIP] {mask mask} {data data} } {actions [{drop | forward } {cos cos} {-1} {mirror-destination control_index} {syslog] }] } {detail}`
15	Shows the pre-defined action set for use in RADIUS Change of Authentication (CoA).	`show policy access-list action-set {set_id}`

Table 101: Setting Up Convergence End Point (CEP) Detection

Step	Task	Command(s)
1	Enable policy globally on the switch.	`enable policy`
2	Enable CEP detection globally on the switch.	`configure policy convergence-endpoint [enable | disable]`

Table 101: Setting Up Convergence End Point (CEP) Detection (continued)

Step	Task	Command(s)
3	Enable CEP detection type on one or more ports.	`configure policy convergence-endpoint ports [<port_list> | all] [cisco | lldp-med] [enable | disable]`
4	Configure a policy to apply to the detected CEP devices.	`configure policy profile profile_index {name name} {pvid pvid} {pvid-status pvid_status} {cos cos} {cos-status cos_status} {egress-vlans egress_vlan_list} {forbidden-vlans forbidden_vlans} {untagged-vlans untagged_vlans} {append | clear} {tci-overwrite tci_overwrite} {precedence [precedence | default]} {auth-override auth_override} {nsi [nsi | none]} {web-redirect web_redir_index} {access-list [unassigned | list_name | list_name_placeholder]}`
5	Assign the configured policy to the desired CEP detection type.	`configure policy convergence-endpoint index index [cisco | lldp-med]`

Table 102: Setting Up Captive Portal Redirection

Step	Task	Command(s)
1	Define a role that has a valid captive portal web redirection class index.	`configure policy profile profile_index {name name} {pvid pvid} {pvid-status pvid_status} {cos cos} {cos-status cos_status} {egress-vlans egress_vlan_list} {forbidden-vlans forbidden_vlans} {untagged-vlans untagged_vlans} {append | clear} {tci-overwrite tci_overwrite}`

Table 102: Setting Up Captive Portal Redirection (continued)

Step	Task	Command(s)
		{**precedence** [*precedence* | **default**]} {**auth-override** *auth_override*} {**nsi** [*nsi* | **none**]} {**web-redirect** *web_redir_index*} {**access-list** [**unassigned** | *list_name* | *list_name_placeholder*]}
2	Configure a captive portal server's HTTP redirect URL and enable it using the previously defined captive portal web redirection class index.	configure policy captive-portal **web-redirect** *redirect_index* **server** *server_id* {**url** *redirect_url*} {*status*}
3	Configure which L4 listening ports (sockets) to be redirected when a captive portal web-redirect is defined on a policy profile.	configure policy captive-portal **listening** *socket_list*
4	Set whether or not captive portal rules are programmed within the already reserved ACL rule space of ONEPolicy.	configure policy **captive-portal rule-use** [**reserved** | **unreserved**]

Table 103: Setting up Policy-Based Mirrors

Step	Task	Command(s)
1	Create a "control group" (Mirror MIB) instance.	create mirror *control_index*
2	Create one or more "physical" mirrors. **Note:** MIB support (not CLI configured) adds a port to a mirror by the ports' Interface Index. In this case, the "physical" mirror is automatically created if there are resources and a matching "physical" mirror does not already exist. To assign a mirror with a remote-ip destination, the mirror must be configured before it can be added to a mirror control index. An interface index for the tunnel mirror is created at the time the tunnel mirror is created. This Interface Index can then be added to the mirror control_index.	create mirror *mirror_name* {**to** [**port** *port* | **port-list** *port_list* **loopback-port** *port*] **{ remote-tag** *rtag* } | {{ } {} { [|]} { [|]} **priority** *priority_value*]} {**description** *mirror-desc*}
3	Apply the physical mirrors to the control group instance using the add option.	configure mirror *control_index* [**add** | **delete**] *mirror_name*

Table 103: Setting up Policy-Based Mirrors (continued)

Step	Task	Command(s)
4	Enable the control group (Mirror MIB) instance, and the physical mirrors.	`enable mirror` *control_index* {**mirror** *mirror_name*}
5	Apply the control group (Mirror MIB) instance to the desired policy using the *control_index* designator.	`configure policy rule` *profile_index* [{**app-signature group** *group* **name** *name*} | **ether** *ether* | **icmp6type** *icmp6type* | **icmptype** *icmptype* | **ip6dest** *ip6dest* |**ipdestsocket** *ipdestsocket* | **ipfrag** | **ipproto** *ipproto* | **ipsourcesocket** *ipsourcesocket* | **iptos** *iptos* | **ipttl** *ipttl* | **macdest** *macdest* | **macsource** *macsource* | **port** *port* | **tcpdestportIP** *tcpdestportIP* | **tcpsourceportIP** *tcpsourceportIP* | **udpdestportIP** *udpdestportIP* | **udpsourceportIP** *udpsourceportIP*] {**mask** *mask* } {**port-string** [*port_string* | **all**]} {**storage-type** [**non-volatile** | **volatile**]} {**drop** | **forward**} {**syslog** *syslog*} {**trap** *trap*} {**cos** *cos* } {**mirror-destination** *control_index*} {**clear-mirror**}

Table 104: Setting up Rule Trap and Syslog

Step	Task	Command(s)
1	Configure a policy rule with a Syslog (syslog option) or trap (trap option) action.	`configure policy rule` *profile_index* [{**app-signature group** *group* **name** *name*} | **ether** *ether* | **icmp6type** *icmp6type* | **icmptype** *icmptype* | **ip6dest** *ip6dest* |**ipdestsocket** *ipdestsocket* | **ipfrag** | **ipproto** *ipproto* | **ipsourcesocket**

Table 104: Setting up Rule Trap and Syslog (continued)

Step	Task	Command(s)
		ipsourcesocket | **iptos** *iptos* | **ipttl** *ipttl* | **macdest** *macdest* | **macsource** *macsource* | **port** *port* | **tcpdestportIP** *tcpdestportIP* | **tcpsourceportIP** *tcpsourceportIP* | **udpdestportIP** *udpdestportIP* | **udpsourceportIP** *udpsourceportIP*] {**mask** *mask* } {**port-string** [*port_string* | **all**]} {**storage-type** [**non-volatile** | **volatile**]} {**drop** | **forward**} {**syslog** *syslog*} {**trap** *trap*} {**cos** *cos* } {**mirror-destination** *control_index*} {**clear-mirror**}
2	Optionally, for Syslog, you can change the following parameters: • Hexadecimal versus decimal format (machine-readable) • Extended format (extended-format) • Syslog message sent only on first rule usage or every time the rule is used (every-time)	configure policy **syslog** [**machine-readable** *machine_readable* | **extended-format** *extended_format* | **every-time** *every_time*]
3	Optionally, set the interval at which the switch automatically clears rule usage statistics.	configure policy **autoclear** {**interval** *interval*}
4	Optionally, clear the policy counters (rule usage) at any time.	clear counters policy

Table 105: Configuring Layer 7 Policy/Application Signatures

Step	Task	Command(s)
1	Configure the shared slice in the switch before enforcing additional configuration.	configure policy **slices shared** [{ *shared* } { **l7GuaranteedPercentage** *l7GuaranteedPercentage* } { **dynAclGuaranteedPercentage**

Table 105: Configuring Layer 7 Policy/Application Signatures (continued)

Step	Task	Command(s)
		dynAclGuaranteedPercentage}]
2	Configure a user-defined policy application signature.	configure policy **app-signature group** *group* **name** *name* [**add** | **delete**] *pattern_list*
3	Apply the signature to a policy rule using the app-signature, group, and name options.	configure policy rule *profile_index* [{**app-signature group** *group* **name** *name*} | **ether** *ether* | **icmp6type** *icmp6type* | **icmptype** *icmptype* | **ip6dest** *ip6dest* |**ipdestsocket** *ipdestsocket* | **ipfrag** | **ipproto** *ipproto* | **ipsourcesocket** *ipsourcesocket* | **iptos** *iptos* | **ipttl** *ipttl* | **macdest** *macdest* | **macsource** *macsource* | **port** *port* | **tcpdestportIP** *tcpdestportIP* | **tcpsourceportIP** *tcpsourceportIP* | **udpdestportIP** *udpdestportIP* | **udpsourceportIP** *udpsourceportIP*] {**mask** *mask* } {**port-string** [*port_string* | **all**]} {**storage-type** [**non-volatile** | **volatile**]} {**drop** | **forward**} {**syslog** *syslog*} {**trap** *trap*} {**cos** *cos* } {**mirror-destination** *control_index*} {**clear-mirror**}
4	Configure the number of slices that a profile with tci-overwrite-enabled uses.	configure policy **slices** {**shared** *shared*} {**tci-overwrite** *slices*}

Table 105: Configuring Layer 7 Policy/Application Signatures (continued)

Step	Task	Command(s)
5	Configure the number of slices used by shared features (Layer 7 policy and dynamic ACLs).	`configure policy` **`slices shared`** `[{` *`shared`* `}` `{` **`l7GuaranteedPercentage`** *`l7GuaranteedPercentage`* `}` `{` **`dynAclGuaranteedPercentage`** *`dynAclGuaranteedPercentage`*`}]`
6	Optionally, configure the minimum time-to-live (TTL).	`configure policy` **`app-signature minimum-ttl`** `[`**`none`** `|` **`1`** `|` **`5`** `|` **`10`**`]`

Table 106: Configuring ACL Style Policy

Step	Task	Command(s)
1	Enable ACL Style Policy. Select the rule model as access-list.	`configure policy` **`rule-model`** `[`**`access-list`** `|` **`hierarchical`**`]`
2	Create a policy access list and match criteria. Access list names are case sensitive. The *`list_dot_rule`* variable cannot match an existing one (using CLI namespace). At least 1, and no more than 5, match criteria can be selected, and at least 1 action must be selected. You cannot edit existing rules. You must delete and re-create rules if you need to make changes and want to use the same *`list_dot_rule`* name. Match conditions must be unique within an ACL and use them as a rule identifier in some cases, such as counters. For example `create policy access-list ACL.ace` produces an error when an ace entry that duplicates an existing "ace" exists in that ACL. Rules with the same match type with different data (IP/Port, etc.) within that the rule are accepted. ACL Style Policy supports the following rule match types: • Application—Layer 7, DNS-based • IP source and/or destination socket—IPv4 address mask • IP fragmentation—indicates packet is fragmented • IP protocol type—UDP/TCP/ICMP	`create policy` **`access-list`** *`list_dot_rule`* **`{matches`** `[` **`{app-signature group`** *`group name`* *`name`*`}` **`{ether`** *`ether`* **`{mask`** *`ether_mask`*`}}` **`{icmp6type`** *`icmp6type`* **`{mask`** *`icmp6_mask`*`}}` **`{icmptype`** *`icmptype`* **`{mask`** *`icmp_mask`*`}}` **`{ipdestsocket`** *`ipdestsocket`* **`{mask`** *`ipdest_mask`*`}}` **`{ipfrag`**`}` **`{ipproto`** *`ipproto`* **`{mask`** *`ipproto_mask`*`}}` **`{ipsourcesocket`** *`ipsourcesocket`* **`{mask`** *`ipsrc_mask`*`}}` **`{iptos`** *`iptos`* **`{mask`** *`iptos_mask`*`}}` **`{ipttl`** *`ipttl`* **`{mask`** *`ipttl_mask`*`}` **`{tcpdestportIP`** *`tcpdestportIP`* **`{mask`** *`tcpdest_mask`*`}}` **`{tcpsourceportIP`** *`tcpsourceportIP`* **`{mask`** *`tcpsrc_mask`*`}}` **`{udpdestportIP`** *`udpdestportIP`* **`{mask`** *`udpdest_mask`*`}}` **`{udpsourceportIP`**

Table 106: Configuring ACL Style Policy (continued)

Step	Task	Command(s)
	• Source/destination UDP or TCP port—L4 port • ICMPv4 or ICMPv6 Type—ICMP packet types • IP TOS • IP TTL • Ether type • ip6dest ACL rules can combine up to 5 match types in a single rule. However, there are a few limitations for specified combinations: • The match type ipfrag cannot be combined with ICMP, TCP, or UDP rules, since the header is only in the first packet. • The match types tcp and udp cannot be in a single rule since you can never have both. • The match types icmp6type and icmptype cannot be combined together. • The match type icmp6type cannot be combined with IPv4 rules. • The match type icmp cannot be combined with tcp or udp rules. • The ether type is limited with compatible types. For example: IPv4 (0x0800) with IPv4 rules (such as icmptype/ipsrc/ipdest), IPV6 (0x86DD) with IPv6 capable rules (such as icmp6Type). However, iptos and ipttl work for both, as do udp and tcp, if they only specify ports, and not IP addresses. • The type ipproto is limited to compatible types. For example: ipproto 1 icmp, 4 ipv4, 6 tcp, 7 udp, 58 icmp6. Actions: When forward/deny is not specified in the rule, the rule action is interpreted as "forward" by default. This is an implicit permit, since the rule is triggered.	*udpsourceportIP* {**mask** *udpsrc_mask*}}] } {**actions** [{**cos** *cos*} {**drop** | **forward**} {**mirror-destination** *control_index*} {**syslog**}]}
3	Add rules and configure the rule precedence list for an access list. ip6dest can only be specified if the ingress slice type supports IPv6. When using multiple ingress slice types (`ipv4-single-ipv6-double` or `ipv4-ipv6-double-separate`), the first and last rule options are not supported. Also, relative rules must be in the same slice type as the rule being configured.	`configure policy` **access-list** [**rule-precedence** [*list_dot_rule* [**after** *member_rule* | **before** *member_rule* | **first** | **last**]]]

Table 106: Configuring ACL Style Policy (continued)

Step	Task	Command(s)
4	Assign access list to a profile using the access-list option. **Note:** Use the unassigned option to remove an access list. You can configure the access list name when configuring the policy profile index and create access list rules for the access list later (step 2). An access list must contain at least one rule and is not active until it is assigned to a profile. Assigning a different access list to the same policy profile index overwrites any previous assignment. An access list cannot be assigned to more than one policy profile index. You must unassigned the access list from the current policy profile index before it can be associated with another policy profile index.	`configure policy profile profile_index {name name} {pvid pvid} {pvid-status pvid_status} {cos cos} {cos-status cos_status} {egress-vlans egress_vlan_list} {forbidden-vlans forbidden_vlans} {untagged-vlans untagged_vlans} {append | clear} {tci-overwrite tci_overwrite} {precedence [precedence | default]} {auth-override auth_override} {nsi [nsi | none]} {web-redirect web_redir_index} {access-list [unassigned | list_name | list_name_placeholder]}`
5	(Optional) Delete policy access lists and/or rules. You can remove a specific rule or remove all the rules from an access list, or remove all access lists and their rules.	`delete policy access-list [all-rules | list_dot_rule]`

ONEPolicy Configuration Examples

The following examples demonstrate how to configure various features of ONEPolicy.

Policy Configuration Example

This section presents a college-based policy configuration example. Figure 168 displays an overview of the policy configuration. This overview display is followed by a complete discussion of the configuration example.

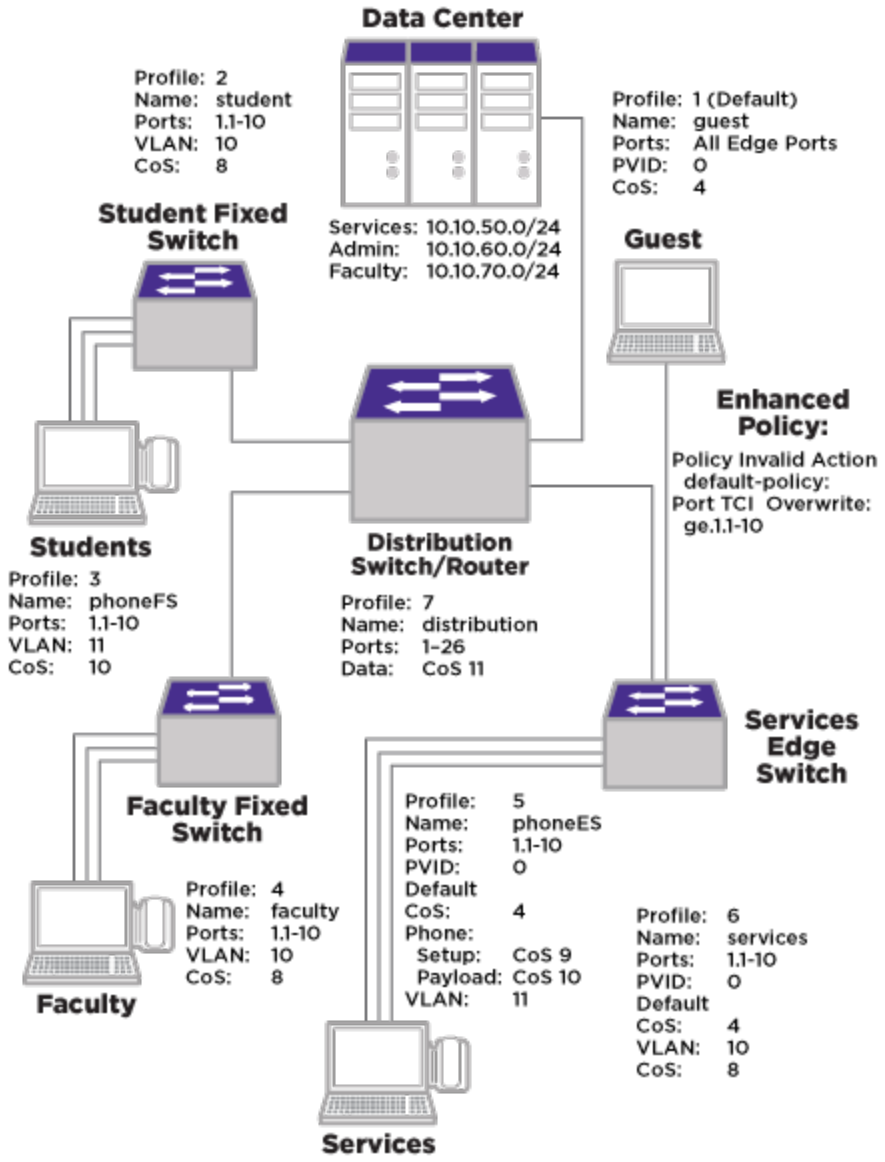


Figure 168: College-Based Policy Configuration

Roles

The example defines the following roles:

- guest – Used as the default policy for all unauthenticated ports. Connects a PC to the network providing internet only access to the network. Provides guest access to a limited number of the edge switch ports to be used specifically for internet only access. Policy is applied using the port level default configuration, or by authentication, in the case of the Services Edge Switch port internet only access PCs.
- student – Connects a dorm room PC to the network through a “Student” Fixed Switch port. A configured *CoS* rate limits the PC. Configured rules deny access to administrative and faculty servers. The PC authenticates using *RADIUS*. Hybrid

authentication is enabled. The student policy role is applied using the filter-ID attribute. The base *VLAN* is applied using the tunnel attributes returned in the RADIUS response message. If all rules are missed, the settings configured in the student policy profile are applied.

- phoneFS – Connects a dorm room or faculty office VoIP phone to the network using a stackable fixed switch port. A configured CoS rate limits the phone and applies a high priority. The phone authenticates using RADIUS. Hybrid authentication is enabled. Policy is applied using the filter-ID returned in the RADIUS response message. The base VLAN is applied using the tunnel attributes returned in the RADIUS response message. If all rules are missed, the settings configured in the phoneFS policy profile are applied.
- faculty – Connects a faculty office PC to the network through a “Faculty” Fixed Switch port. A configured CoS rate limits the PC. A configured rule denies access to the administrative servers. The PC authenticates using RADIUS. Hybrid authentication is enabled. The faculty policy role is applied using the filter-ID attribute. The base VLAN is applied using the tunnel attributes returned in the RADIUS response message for the authenticating user. If all rules are missed, the settings configured in the faculty policy profile are applied.
- phoneES – Connects a services VoIP phone to the network using a Services Edge Switch port. A configured CoS rate limits the phone for both setup and payload, and applies a high priority. The phone authenticates using RADIUS. Tunnel authentication is enabled. The base VLAN is applied using the tunnel attributes returned in the RADIUS response message. Policy is applied using a maptable configuration. If all rules are missed, the settings configured in the phoneES policy profile are applied.
- services – Connects a services PC to the network through the Services Edge Switch port. A configured CoS rate limits the PC. Services are denied access to both the student and faculty servers. The PC authenticates using RADIUS. The base VLAN is applied using the tunnel attributes returned in the RADIUS response message for the authenticating user. The services policy role is applied using a policy maptable setting. The policy invalid action and TCI overwrite are enabled for this role. If all rules are missed, the settings configured in the services policy profile are applied.
- distribution – The Distribution policy role is applied at the Distribution Switch providing rate limiting.

Policy Domains

It is useful to break up policy implementation into logical domains for ease of understanding and configuration. For this example, it is useful to consider four domains: basic edge, standard edge on the Fixed Switch, premium edge on the Services Edge Switch, and premium distribution on the Distribution Switch.

Basic Edge

Protocols not appropriate to the edge should be blocked. For this example we will block *DHCP*, DNS, *SNMP*, SSH, Telnet and FTP at the edge on the data *VLAN*. We will forward destination port DHCP and DNS and source port for IP address request to facilitate auto configuration and IP address assignment. See Blocking Non-Edge Protocols at the Edge Network Layer on page 1347 for a listing of protocols you should consider blocking at the edge.

Standard Edge

Edge Switch platforms will be rate-limited using a configured *CoS* that will be applied to the student and faculty, and phoneFS policy roles. Fixed Switch support for hybrid authentication depends upon the platform and firmware release. The Fixed Switch in this example supports the hybrid authentication capability. Hybrid authentication will be enabled.

Premium Edge

The Edge Switch will be rate-limited using a configured CoS that is applied to the services and phoneES policy role. This premium edge platform will be enabled for the following capabilities:

- Invalid policy action set to drop
- TCI overwrite enabled

Premium Distribution

The Distribution Switch Router will be rate-limited using a configured CoS. Premium distribution will be enabled for the following policy capabilities:

- Invalid policy action set to drop
- TCI overwrite enabled

Platform Configuration

This section provides the CLI based policy configuration on the following platforms:

- Student Fixed Switch
- Faculty Fixed Switch
- Services Edge Switch
- Distribution Switch

In CLI mode, configuration takes place on each platform. For this configuration example, *CoS* related configuration will be specified as a final CoS. See the *QoS* Configuration feature guide located at http://documentation.extremenetworks.com for a complete discussion of QoS configuration.

Note

CLI command prompts used in this configuration example have the following meaning:

- `System->`—Input on all platforms used in this example.
- `Fixed Switch->`—Input on all Fixed Switches.
- `StudentFS->`—Input on the student Fixed Switch.
- `FacultyFS->`—Input on the faculty Fixed Switch.
- `Services->`—Input on the services S-Series device.
- `Distribution->`—Input on the distribution S-Series device.

Configuring Guest Policy on Edge Platforms

All edge ports will be set with a default guest policy using the `configure policy port` command. This guest policy provides for an internet only access to the network. Users on all ports will attempt to authenticate. If the authentication succeeds, the policy returned by authentication or, in the case of the Services Edge Switch configuration, the maptable setting, overrides the default port policy setting. If authentication fails, the guest policy is used. On the Services Edge Switch , five ports are used by PCs at locations throughout the campus, such as the library, to provide access to the internet. The PCs attached to these five ports will authenticate with the guest policy role. Public facing services would also be configured for guest status in a school or enterprise scenario. Public facing services are not part of this example.

Configuring the Policy Role

The guest role is configured with:

- A profile-index value of 1
- A name of guest
- A PVID set to 0
- A *CoS* set to 4

Create the guest policy profile on all platforms:

```
System ->configure policy profile 1 name guest pvid-status enable pvid 0 cos-status
enable cos 4
```

Assigning Traffic Classification Rules

For cases where discovery must take place to assign an IP address, DNS and *DHCP* traffic must be allowed. Forwarding of traffic is allowed on UDP source port 68 (IP address request) and UDP destination ports 53 (DNS) and 67 (DHCP).

```
System ->configure policy rule 1 udpsourceport 68 mask 16 forward
System->configure policy rule 1 udpdestportIP 53 mask 16 forward
System ->configure policy rule 1 udpdestportIP 67 mask 16 forward
```

Guest policy allows internet traffic. TCP destination Ports 80, 8080, and 443 will be allowed traffic forwarding.

```
System->configure policy rule 1 tcpdestportIP 80 mask 16 forward
System->configure policy rule 1 tcpdestportIP 443 mask 16 forward
System->configure policy rule 1 tcpdestport 8080 mask 16 forward
```

ARP forwarding is required on ether port 0x806.

```
System->configure policy rule 1 ether 0x806 mask 16 forward
```

Assigning the Guest Policy Profile to All Edge Ports

Assign the guest policy profile to all Fixed Switch and Services Edge Switch ports.

```
System->configure policy port 1-47 1
```

Configuring Policy for the Edge Student Fixed Switch

Configuring the Policy Role

The student role is configured with:

- A profile-index value of 2
- A name of student
- A port *VLAN* of 10
- A *CoS* of 8

Create a policy role that applies a CoS 8 to data VLAN 10 and configures it to rate-limit traffic to 1M with a moderate priority of 5.

```
StudentFS->configure policy profile 2 name student pvid-status
enable pvid 10 cos-status enable cos 8
```

Assigning Hybrid Authentication

Configure the *RADIUS* server user accounts with the appropriate tunnel information using VLAN authorization and policy filter-ID for student role members and devices. Enable hybrid authentication, allowing the switch to use both the filter-ID and tunnel attributes in the RADIUS response message. Set a VLAN-to-policy mapping as backup incase the response does not include the RADIUS filter-ID attribute. This mapping is ignored in case RADIUS filter-ID attribute is present in the RADIUS response message.

```
StudentFS->configure policy maptable response both
StudentFS->configure policy maptable 10 2
```

Assigning Traffic Classification Rules

Forward traffic on UDP source port for IP address request (68), and UDP destination ports for protocols *DHCP* (67) and DNS (53). Drop traffic on UDP source ports for protocols DHCP (67) and DNS (53). Drop traffic for protocols *SNMP* (161), SSH (22), Telnet (23) and FTP (20 and 21) on both the data and phone VLANs.

```
StudentFS->configure policy rule 2 udpsourceport 68 mask 16 forward
StudentFS->configure policy rule 2 udpdestport 67 mask 16 forward
StudentFS->configure policy rule 2 udpdestport 53 mask 16 forward
StudentFS->configure policy rule 2 udpsourceportIP 67 mask 16 drop
StudentFS->configure policy rule 2 udpsourceportIP 53 mask 16 drop
StudentFS->configure policy rule 2 udpdestportIP 16 mask 16 drop
StudentFS->configure policy rule 2 tcpdestportIP 22 mask 16 drop
StudentFS->configure policy rule 2 tcpdestportIP 23 mask 16 drop
StudentFS->configure policy rule 2 tcpdestportIP 20 mask 16 drop
StudentFS->configure policy rule 2 tcpdestportIP 21 mask 16 drop
```

Students should only be allowed access to the services server (subnet 10.10.50.0/24) and should be denied access to both the administrative (subnet 10.10.60.0/24) and faculty servers (subnet 10.10.70.0/24).

```
StudentFS->configure policy rule 2 ipdestsocket 10.10.60.0 mask 24 drop
StudentFS->configure policy rule 2 ipdestsocket 10.10.70.0 mask 24 drop
```

Configuring PhoneFS Policy for the Edge Fixed Switch

Configuring the Policy Role

The phoneFS role is configured on both the dorm room and faculty office Fixed Switches with:

- A profile-index of 3
- A name of phoneFS
- A port *VLAN* of 11
- A *CoS* of 10

Because we can not apply separate rate limits to the phone setup and payload ports on the Fixed Switch using policy rules, apply CoS 10 with the higher payload appropriate rate limit of 100k bps and a high priority of 6 to the phoneFS role.

```
Fixed Switch->configure policy profile 3 name phoneFS pvid-status enable pvid 11
cos-status enable cos 10
```

Assigning Traffic Classification Rules

Drop traffic for protocols *SNMP* (161), SSH (22), Telnet (23) and FTP (20 and 21) on the phone VLAN. Forward traffic on UDP source port for IP address request (68) and forward traffic on UDP destination ports for protocols *DHCP* (67) and DNS (53) on the phone VLAN, to facilitate phone auto configuration and IP address assignment.

```
Fixed Switch->configure policy rule 3 udpdestportIP 161 mask 16 drop
Fixed Switch->configure policy rule 3 tcpdestportIP 22 mask 16 drop
Fixed Switch->configure policy rule 3 tcpdestportIP 23 mask 16 drop
Fixed Switch->configure policy rule 3 tcpdestportIP 20 mask 16 drop
Fixed Switch->configure policy rule 3 tcpdestportIP 21 mask 16 drop
Fixed Switch->configure policy rule 3 udpsourceport 68 mask 16 forward
Fixed Switch->configure policy rule 3 udpdestportIP 67 mask 16 forward
Fixed Switch->configure policy rule 3 udpdestportIP 53 mask 16 forward
```

Assigning Hybrid Authentication

Configure the *RADIUS* server user accounts with the appropriate tunnel information using VLAN authorization and policy filter-ID for phoneFS role members and devices. Enable hybrid authentication, allowing the switch to use both the filter-ID and tunnel attributes in the RADIUS response message. Set a VLAN-to-policy mapping as backup incase the response does not include the RADIUS filter-ID attribute. This mapping is ignored if RADIUS filter-ID attribute is present in the RADIUS response message.

```
Fixed Switch->configure policy maptable response both
Fixed Switch->configure policy maptable 11 3
```

Configuring Policy for the Edge Faculty Fixed Switch

Configuring the Policy Role

The faculty role is configured with:

- A profile-index value of 4
- A name of faculty
- A port *VLAN* of 10
- A *CoS* of 8

Create a policy role that applies a CoS 8 to data VLAN 10 and configures it to rate-limit traffic to 1M with a moderate priority of 5.

```
FacultyFS->configure policy profile 4 name faculty pvid-status enable pvid 10
cos-status enable cos 8
```

Assigning Hybrid Authentication

Configure the *RADIUS* server user accounts with the appropriate tunnel information using VLAN authorization and policy filter-ID for faculty role members and devices. Enable hybrid authentication. Set a VLAN-to-policy mapping. This mapping is ignored if the RADIUS filter-ID attribute is present in the RADIUS response message.

```
FacultyFS->configure policy maptable response both
FacultyFS->configure policy maptable 10 4
```

Assigning Traffic Classification Rules

Forward traffic on UDP source port for IP address request (68), and UDP destination ports for protocols *DHCP* (67) and DNS (53). Drop traffic on UDP source ports for protocols DHCP (67) and DNS (53). Drop traffic for protocols *SNMP* (161), SSH (22), Telnet (23) and FTP (20 and 21) on both the data and phone VLANs

```
FacultyFS->configure policy rule 4 udpsourceport 68 mask 16 forward
FacultyFS->configure policy rule 4 udpdestport 67 mask 16 forward
FacultyFS->configure policy rule 4 udpdestport 53 mask 16 forward
FacultyFS->configure policy rule 4 udpsourceportIP 67 mask 16 drop
FacultyFS->configure policy rule 4 udpsourceportIP 53 mask 16 drop
FacultyFS->configure policy rule 4 udpdestportIP 16 mask 16 drop
FacultyFS->configure policy rule 4 tcpdestportIP 22 mask 16 drop
FacultyFS->configure policy rule 4 tcpdestportIP 23 mask 16 drop
FacultyFS->configure policy rule 4 tcpdestportIP 20 mask 16 drop
FacultyFS->configure policy rule 4 tcpdestportIP 21 mask 16 drop
```

Faculty should only be allowed access to the services (subnet 10.10.50.0/24) and the faculty servers (subnet 10.10.70.0/24) and should be denied access to the administrative server (subnet 10.10.60.0/24).

```
FacultyFS->configure policy rule 4 ipdestsocket 10.10.60.0 mask 24 drop
```

Configuring PhoneES Policy for the Services Edge Switch

Configuring the Policy Role

The phoneES role is configured on the Services Edge Switch with:

- A profile-index of 5 Policy
- A name of phoneES
- A default port *VLAN* of 0
- A default *CoS* of 4

Because VLANs can be applied to Services Edge Switch ports using the appropriate traffic classification, the explicit deny all PVID 0 will be applied at policy creation. Separate rate limits can be applied to the phone setup and payload ports on the

Services Edge Switch using policy rules. A default CoS of 4 will be applied at policy role creation.

```
ServicesES->configure policy profile 5 name phoneES pvid-status enable pvid 0
cos-status enable cos 4
```

Assigning Traffic Classification Rules

Forward traffic on UDP source port for IP address request (68) and and forward traffic on UDP destination ports for protocols *DHCP* (67) and DNS (53) on the phone VLAN, to facilitate phone auto configuration and IP address assignment. Drop traffic for protocols *SNMP* (161), SSH (22), Telnet (23) and FTP (20 and 21) on the phone VLAN.

```
ServicesES->configure policy rule 5 udpsourceport 68 mask 16 forward
ServicesES->configure policy rule 5 udpdestportIP 67 mask 16 forward
ServicesES->configure policy rule 5 udpdestportIP 53 mask 16 forward
ServicesES->configure policy rule 5 udpdestportIP 161 mask 16 drop
ServicesES->configure policy rule 5 tcpdestportIP 22 mask 16 drop
ServicesES->configure policy rule 5 tcpdestportIP 23 mask 16 drop
ServicesES->configure policy rule 5 tcpdestportIP 20 mask 16 drop
ServicesES->configure policy rule 5 tcpdestportIP 21 mask 16 drop
```

Apply a CoS 9 to phone setup data on VLAN 11, rate limiting the data to 5 pps with a high priority of 7 on port 2427.

Apply a CoS 10 to phone payload data on VLAN 11, rate limiting the data to 100k bps with a high priority of 7 for both source and destination on port 5004.

```
ServicesES->configure policy rule 5 udpdestIP 2427 mask 16 vlan 11 cos 9
ServicesES->configure policy rule 5 udpsourceIP 5004 mask 16 vlan 11 cos 10
ServicesES->configure policy rule 5 udpdestIP 5004 mask 16 vlan 11 cos 10
```

Assigning the VLAN-to-Policy Association

The nature of services related devices that might connect to a switch port is not as static as with the student or faculty roles. Services related network needs can run the gamut from temporary multimedia events to standard office users. There may be multiple VLAN and policy role associations that take care of services related needs, depending upon the connected user. This may include the requirement for multiple services related roles.

For services, the network administrator desires greater resource usage flexibility in assigning the policy to VLAN association. Authentication in this case will return only the tunnel attributes in the response message based upon the requirements of the authenticating user. Setting the VLAN-to-policy association will be handled by the maptable configuration, allowing for ease in changing the policy associated with a VLAN on the fly using Policy Manager. Specify that the tunnel attributes returned in the *RADIUS* response message will be used by the authenticating user. Associate VLAN 11 with policy role 5 using the `configure policy maptable` command.

```
ServicesES->configure policy maptable response tunnel
ServicesES->configure policy maptable 11 5
```

Configuring Policy for the Services Edge Switch

Configuring the Policy Role

The services role is configured with:

- A profile-index value of 6
- A name of services
- A default port *VLAN* of 0
- A default *CoS* when no rule overrides CoS
- TCI overwrite enabled

```
ServicesES->configure policy profile 6 name services pvid-status enable pvid 0
cos-status enable cos 4 tci-overwrite enable
```

Assigning the VLAN-to-Policy Association

Setting the VLAN-to-policy association will be handled by the policy maptable setting, allowing for ease in changing the policy associated with a VLAN on the fly using Policy Manager. Specify that the tunnel attributes returned in the *RADIUS* response message will be used by the authenticating user. Associate VLAN 10 with policy role 6 using the `set policy maptable` command.

```
ServicesES->configure policy maptable response tunnel
ServicesES->configure policy maptable 10 6
```

Assigning Traffic Classification Rules

Forward traffic on UDP source port for IP address request (68) and forward traffic on UDP destination ports for protocols *DHCP* (67) and DNS (53) on the data VLAN, to facilitate PC auto configuration and IP address assignment. Drop traffic for protocols *SNMP* (161), SSH (22), Telnet (23) and FTP (20 and 21) on the phone VLAN.

```
ServicesES->configure policy rule 6 udpsourceportIP 68 mask 16 vlan 10 forward
ServicesES->configure policy rule 6 udpdestportIP 67 mask 16 vlan 10 forward
ServicesES->configure policy rule 6 udpdestportIP 53 mask 16 vlan 10 forward
ServicesES->configure policy rule 6 udpdestportIP 67 mask 16 vlan 10 drop
ServicesES->configure policy rule 6 udpdestportIP 53 mask 16 vlan 10 drop
ServicesES->configure policy rule 6 udpdestportIP 161 mask 16 drop
ServicesES->configure policy rule 6 tcpdestportIP 22 mask 16 drop
ServicesES->configure policy rule 6 tcpdestportIP 23 mask 16 drop
ServicesES->configure policy rule 6 tcpdestportIP 20 mask 16 drop
ServicesES->configure policy rule 6 tcpdestportIP 21 mask 16 drop
```

Apply a CoS 8 to data VLAN 10 and configure it to rate-limit traffic to 1M and moderate priority of 5 for services IP subnet 10.10.30.0 mask 28.

```
ServicesES->configure policy rule 6 ipsourcesocket 10.10.30.0 mask 28 vlan 10 cos 8
```

Services should only be allowed access to the services server (subnet 10.10.50.0/24) and should be denied access to the faculty servers (subnet 10.10.70.0/24) and administrative servers (subnet 10.10.60.0/24).

```
ServicesES->configure policy rule 6 ipdestsocket 10.10.60.0 mask 24 drop
ServicesES->configure policy rule 6 ipdestsocket 10.10.70.0 mask 24 drop
```

Enable Enhanced Edge Switch Capabilities on the Services Edge Switch Platform

The Services Edge Switch platform supports invalid action set to default policy should an invalid policy occur.

```
ServicesES->configure policy invalid action default-policy
```

Configuring the Distribution Layer Role

Configuring the Policy Role

The distribution role is configured with:

- A profile-index value of 7
- A name of distribution
- A default *CoS* when no rule overrides CoS
- TCI overwrite enabled

```
Distribution->configure policy profile 7 name distribution cos-status enable cos 4
tci-overwrite enable
```

Assigning Traffic Classification Rules

Assign a CoS to distribution up and down stream link ports, rate-limiting the traffic to 25M.

```
Distribution->configure policy rule 7 port 1-26 cos 11
Distribution->configure policy rule 7 port 1-26 cos 11
```

Enable Enhanced Policy Capability

The following enhanced policy capability is enabled: invalid action set to default policy should an invalid policy occur.

```
ServicesES->configure policy invalid action default-policy
```

This completes the policy configuration for this school example.

Captive Portal Redirection Example

In the following example:

- Switch is configured to listen for web traffic on ports 80 and 8080.
- Web-redirect index 1 enables server 1 with a redirection URL http://10.52.3.101:80/static/index.jsp.
- Web-redirect index 1 enables server 2 with a redirection URL http://10.52.3.102:80/static/index.jsp.
- Web-redirect index 1 is associated with policy profile 1.

```
#configure policy captive-portal listening 80,8080
#configure policy captive-portal web-redirect 1 server 1 url http:// 10.52.3.101:80/
static/index.jsp enable
#configure policy captive-portal web-redirect 1 server 2 url http:// 10.52.3.102:80/
static/index.jsp enable
#configure policy profile 1 web-redirect 1

#show policy captive-portal listening
```

```
Captive Portal Listening Ports: 80 8080

#show policy captive-portal web-redirect 1
Web-redirect Index: 1
Server Index: 1
Server Status: Enabled
Server URL: http://10.52.3.101:80/static/index.jsp
Server Index: 2
Server Status: Enabled
Server URL: http://10.52.3.102:80/static/index.jsp
```

Sending a Change of Authentication (CoA) from FreeRADIUS Example

To add a DACL:

```
Example: echo "Calling-Station-ID = <SESSION-MAC-HYPONATED>, NAS-IP-Address = <switch ip-
address>, Tunnel-Medium-Type = IEEE-802, Tunnel-Type = VLAN, Tunnel-Private-Group-ID =
<VLAN-ID/STRING>, Filter-Id = <Profile-Name>, Extreme-Policy-ACL = \"v:1 t:a
m:ipv4src=10.10.10.1/32,ipv4dst=192.168.10.25/32,ipproto=tcp,l4srcport=69/8,l4dstport=69/8
 a:fwd,sys,cos=2 \" " | radclient -x -r 1 <Switch-IP>:3799 coa secret
```

To delete the DACL:

```
echo "Calling-Station-ID = <SESSION-MAC-HYPONATED>, NAS-IP-Address = <switch ip-address>,
Tunnel-Medium-Type = IEEE-802, Tunnel-Type = VLAN, Tunnel-Private-Group-ID = <VLAN-ID/
STRING>, Filter-Id = <Profile-Name>, Extreme-Policy-ACL = \"v:1 t:d
m:ipv4src=10.10.10.1/32,ipv4dst=192.168.10.25/32,ipproto=tcp,l4srcport=69/8,l4dstport=69/8
 a:fwd,sys,cos=2 \" " | radclient -x -r 1 <Switch-IP>:3799 coa secret
```

Example Dynamic ACL VSA String

This example shows adding two rules as part of Access-Accept operation.

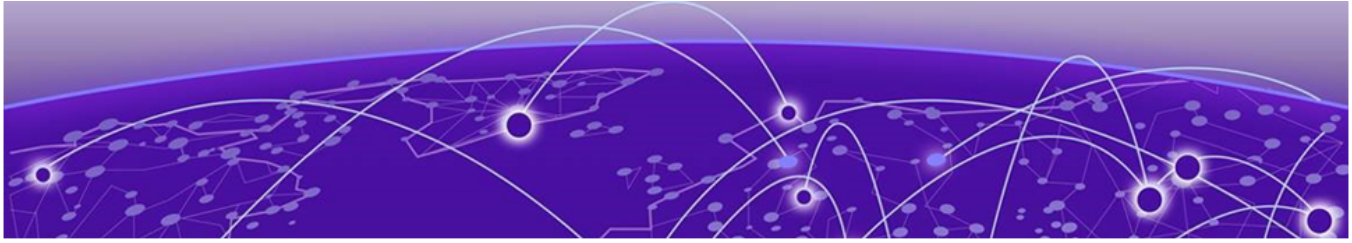
```
Extreme-Policy-ACL = "v:1 t:a
m:ipv4src=1.2.3.4/32,ipv4dst=2.4.6.8/32,ipproto=tcp,l4srcport=4321/16,l4dstport=9876/16
a:fwd,sys,cos=4,mir=2"
```

```
Extreme-Policy-ACL += "v:1 t:a
m:ipv4src=1.2.3.6/32,ipv4dst=2.4.6.8/32,ipproto=tcp,l4srcport=4321/16,l4dstport=9876/16
a:fwd,sys,cos=4,mir=2"
```

```
Extreme-Policy-ACL += "v:1 t:a,r
m:ipv4src=1.2.3.6/32,ipv4dst=2.4.6.8/32,ipproto=tcp,l4srcport=101-200,l4dstport=201-300
a:fwd,sys,cos=4,mir=2"
```

This example shows the application of dacl in multiauth, if the dot1x session needs coa, echo:

```
Calling-Station-ID = <session-mac-hyponated>, NAS-IP-Address = <switch-ip>, Tunnel-Medium-
Type = IEEE-802, Tunnel-Type = VLAN, Tunnel-Private-Group-ID = 101, Enterasys-Auth-Client-
Type = 802.1X, Filter-Id = Default" | radclient -x -r 1 <switch-ip>:3799 coa secret
```



OSPF

OSPF Overview on page 1391
Route Redistribution on page 1401
Configuring OSPF on page 1405
OSPF Configuration Example on page 1406
Displaying OSPF Settings on page 1409

This chapter discusses the OSPF (Open Shortest Path First) protocol for distributing routing information between routers belonging to an autonomous system. This chapter provides an overview of the protocol's features and example configuration commands.

This chapter assumes that you are already familiar with IP unicast routing. If not, refer to the following publications for additional information:

- RFC 2328—OSPF Version 2
- RFC 1765—OSPF Database Overflow
- RFC 2370—The OSPF Opaque LSA Option
- RFC 3101—The OSPF Not-So-Stubby Area (NSSA) Option
- RFC 3623—Graceful OSPF Restart
- *Interconnections: Bridges and Routers* by Radia Perlman. Published by Addison-Wesley Publishing Company (ISBN 0-201-56332-0).

Note
OSPF is available on platforms with an Advanced Edge (or higher) or Base (or higher) license. For specific information regarding OSPF licensing, see the licensing guide.

OSPF Overview

OSPF (Open Shortest Path First) is a link state protocol that distributes routing information between routers belonging to a single IP domain; the IP domain is also known as an autonomous system (AS).

In a link-state routing protocol, each router maintains a database describing the topology of the AS. Each participating router has an identical database maintained from the perspective of that router.

From the LSDB (link state database), each router constructs a tree of shortest paths, using itself as the root. The shortest path tree provides the route to each destination in

the AS. When several equal-cost routes to a destination exist, traffic can be distributed among them. The cost of a route is described by a single metric.

OSPF is an IGP (interior gateway protocol), as is *RIP (Routing Information Protocol)*, the other common IGP. OSPF and RIP are compared in RIP.

Note
Two types of OSPF functionality are available and each has a different licensing requirement. One is the complete OSPF functionality and the other is OSPF Edge Mode, a subset of OSPF that is described below. For specific information regarding OSPF licensing, see the Switch Engine v33.6.1 Licensing Guide document.

OSPF Edge Mode

OSPF Edge Mode is a subset of OSPF available on platforms with an Advanced Edge license.

There are two restrictions on OSPF Edge Mode:

- At most, four Active OSPF *VLAN (Virtual LAN)* interfaces are permitted. There is no restriction on the number of Passive interfaces.
- The OSPF Priority on VLANs is 0, and is not configurable. This prevents the system from acting as a DR or BDR.

BFD for OSPF

The BFD for *OSPF* feature gives OSPF routing protocol the ability to utilize BFD's fast failure detection to monitor OSPF neighbor adjacencies. CLI commands are provided to configure BFD protection for OSPF, so that as a registered BFD client, OSPF can request BFD protection for interested OSPF neighbors, and receive notifications about BFD session setup status and BFD session status updates (after establishing) from the BFD server. When BFD detects a communication failure between neighbors, it informs OSPF, which causes the OSPF neighbor state be marked as "down." This allows OSPF protocol to quickly begin network convergence and use alternate paths to the affected neighbor.

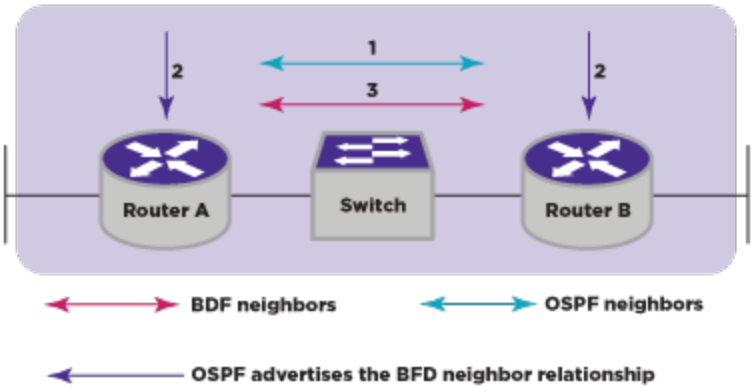


Figure 169: Basic Operational Flow of OSPF BFD

Establishing a BFD Session for OSPF Neighbor

1. OSPF discovers a neighbor.
2. If BFD for OSPF is configured, OSPF on both routers sends a request to the local BFD server to initiate a BFD neighbor session with the OSPF neighbor router.
3. The BFD neighbor session with the OSPF neighbor router is established on both sides if BFD session limit is not reached.

If the BFD session limit is reached, the OSPF neighbor will be marked as BFD session failed if synchronous request is used, or pending if asynchronous request is used, and the BFD server will send an asynchronous notification when the session registration passes later. (The asynchronous request is not available until the BFD client session create API is enhanced.)

Eliminating the OSPF Neighbor Relationship by BFD Fault Detection

1. A failure occurs in the network.
2. The BFD neighbor session with the OSPF neighbor router is removed because the BFD timer expired.
3. On both Router A and Router B, BFD notifies the local OSPF process that the BFD neighbor is DOWN.
4. The local OSPF process tears down the OSPF neighbor relationship by marking neighbor state DOWN. (If an alternative path is available, the routers will immediately start converging on it.)

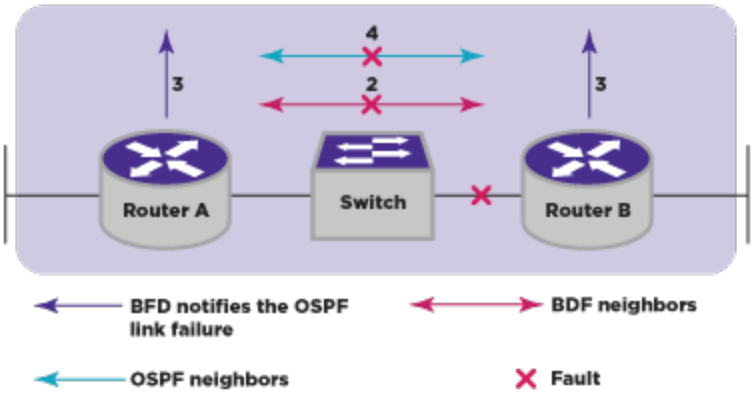


Figure 170: OSPF Neighbor Relationship Eliminated by BFD Fault Detection

OSPF will request the BFD server to delete the BFD session for OSPF neighbor moved to DOWN state. When the link failure is later resolved, OSPF needs to register the re-discovered neighbor to BFD again to initiate BFD session creation.

Removing BFD Protection for OSPF

1. BFD protection is removed from OSPF interface on Router A, OSPF process requests BFD server to delete all BFD sessions for neighbors learned on that interface.
2. BFD server will stop sending session status update to local OSPF process on Router A.
3. Before actually deleting any sessions, BFD server on Router A will first notify Router B to mark those session status as "Admin Down," which will cause Router B to stop using BFD protection for those OSPF neighbors.
4. When BFD protection is removed from OSPF interface on Router B, BFD sessions can be deleted immediately since they are already in "Admin Down" state.

When BFD for OSPF is configured on broadcast interface, the default behavior is to register only OSPF neighbors in FULL state with the BFD server. Separate BFD sessions are created for each neighbor learned on the same interface. If multiple clients ask for the same neighbor on the same interface, then single BFD sessions are established between the peers.

OSPF Neighbor State Determination

With active BFD protection, OSPF combines the BFD session state with the associated interface admin and operational states to determine the OSPF neighbor adjacency discovered on that OSPF interface. Regarding OSPF neighbor relationships, OSPF reacts directly to BFD session state changes only in the following circumstances:

- If BFD is enabled on the interface, and
- If BFD for OSPF is configured on the OSPF interface, and
- If a BFD session has been established to the neighbor, and
- If the BFD session has passed the INIT_COMPLETE state then:

1. The OSPF neighbor relationship will remain as FULL if the operational state of the BFD session is "UP" and the operational state of the associated *VLAN* interface is UP.

2. The OSPF neighbor relationship will be considered as DOWN if the operational state of the BFD session is DOWN or the operational state of the associated VLAN interface is DOWN.

In all other cases, the BFD session state is not considered as part of the reported OSPF neighbor state, and the OSPF neighbor state reverts to the operational state of the OSPF interface only. When the BFD session is in ADMIN_DOWN state, OSPF ignores BFD events and OSPF neighbor adjacency is not be affected by the BFD session state change.

Link State Database

Upon initialization, each router transmits a link state advertisement (LSA) on each of its interfaces. LSAs are collected by each router and entered into the LSDB of each router. After all LSAs are received, the router uses the link state database (LSDB) to calculate the best routes for use in the IP routing table. *OSPF* uses flooding to distribute LSAs between routers. Any change in routing information is sent to all of the routers in the network. All routers within an area have the exact same LSDB.

Table 107 describes LSA type numbers.

Table 107: LSA Type Numbers

Type Number	Description
1	Router LSA
2	Network LSA
3	Summary LSA
4	AS summary LSA
5	AS external LSA
7	NSSA external LSA
9	Link local—Opaque
10	Area scoping—Opaque
11	AS scoping—Opaque

Database Overflow

The *OSPF* database overflow feature allows you to limit the size of the LSDB and to maintain a consistent LSDB across all the routers in the domain, which ensures that all routers have a consistent view of the network.

Consistency is achieved by:

- Limiting the number of external LSAs in the database of each router.
- Ensuring that all routers have identical LSAs.
- To configure OSPF database overflow, use the following command:

  `configure ospf` **`ase-limit`** *`number`* {**`timeout`** *`seconds`*}

Where:

- *number*—Specifies the number of external LSAs that the system supports before it goes into overflow state. A limit value of 0 disables the functionality. When the LSDB size limit is reached, OSPF database overflow flushes LSAs from the LSDB. OSPF database overflow flushes the same LSAs from all the routers, which maintains consistency.
- timeout—Specifies the timeout, in seconds, after which the system ceases to be in overflow state. A timeout value of 0 leaves the system in overflow state until OSPF is disabled and re-enabled.

Opaque LSAs

Opaque LSAs are a generic *OSPF* mechanism used to carry auxiliary information in the OSPF database. Opaque LSAs are most commonly used to support OSPF traffic engineering.

Normally, support for opaque LSAs is autonegotiated between OSPF neighbors.

- In the event that you experience interoperability problems, you can disable opaque LSAs across the entire system using the following command:
  ```
  disable ospf capability opaque-lsa
  ```
- Re-enable opaque LSAs across the entire system:
  ```
  enable ospf capability opaque-lsa
  ```
  If your network uses opaque LSAs, we recommend that all routers on your OSPF network support opaque LSAs. Routers that do not support opaque LSAs do not store or flood them. At minimum a well interconnected subsection of your OSPF network must support opaque LSAs to maintain reliability of their transmission.

Graceful OSPF Restart

RFC 3623 describes a way for *OSPF* control functions to restart without disrupting traffic forwarding.

Without graceful restart, adjacent routers will assume that information previously received from the restarting router is stale and will not be used to forward traffic to that router. However, in many cases, two conditions exist that allow the router restarting OSPF to continue to forward traffic correctly. The first condition is that forwarding can continue while the control function is restarted. Most modern router system designs separate the forwarding function from the control function so that traffic can still be forwarded independent of the state of the OSPF function. Routes learned through OSPF remain in the routing table and packets continue to be forwarded. The second condition required for graceful restart is that the network remain stable during the restart period. If the network topology is not changing, the current routing table remains correct. Often, networks can remain stable during the time for restarting OSPF.

Restarting and Helper Mode

Routers involved with graceful restart fill one of two roles: the restarting router or the helper router.

With graceful restart, the router that is restarting sends out Grace-LSAs informing its neighbors that it is in graceful restart mode, how long the helper router should assist

with the restart (the grace period), and why the restart occurred. If the neighboring routers are configured to help with the graceful restart (helper-mode), they will continue to advertise the restarting router as if it was fully adjacent. Traffic continues to be routed as though the restarting router is fully functional. If the network topology changes, the helper routers will stop advertising the restarting router. The helper router will continue in helper mode until the restarting router indicates successful termination of graceful restart, the Grace-LSAs expire, or the network topology changes. A router can be configured for graceful restart, and for helper-mode separately. A router can be a helper when its neighbor restarts, and can in turn be helped by a neighbor if it restarts.

Planned and Unplanned Restarts

Two types of graceful restarts are defined: planned and unplanned.

A planned restart would occur if the software module for *OSPFv3 (Open Shortest Path First version 3)* was upgraded, or if the router operator decided to restart the OSPFv3 control function for some reason. The router has advance warning, and is able to inform its neighbors in advance that OSPFv3 is restarting.

An unplanned restart would occur if there was some kind of system failure that caused a remote reboot or a crash of OSPFv3, or a failover occurs. As OSPFv3 restarts, it informs its neighbors that it is in the midst of an unplanned restart.

You can decide to configure a router to enter graceful restart for only planned restarts, for only unplanned restarts, or for both. Also, you can separately decide to configure a router to be a helper for only planned, only unplanned, or for both kinds of restarts.

Configuring Graceful OSPF Restart

- Configure a router to perform graceful *OSPF* restart:

  ```
  configure ospf restart [none | planned | unplanned | both ]
  ```

  Since a router can act as a restart helper router to multiple neighbors, you will specify which neighbors to help.
- Configure a router to act as a graceful OSPF restart helper:

  ```
  configure ospf [vlan [all | vlan-name] | area area-identifier |
  virtual-link router-identifier area-identifier] restart-helper [none |
  planned | unplanned | both]
  ```
- The graceful restart period sent out to helper routers can be configured with the following command:

  ```
  configure ospf restart grace-period seconds
  ```

  By default, a helper router will terminate graceful restart if received LSAs would affect the restarting router.

This will occur when the restart-helper receives an LSA that will be flooded to the restarting router or when there is a changed LSA on the restarting router's retransmission list when graceful restart is initiated.

- Disable this behavior:

  ```
  disable ospf [vlan [all {vr vrf_name}| vlan-name] | area area-identifier | virtual-link router-identifier area-identifier] restart-helper-lsa-check
  ```

Areas

OSPFv3 allows parts of a network to be grouped together into areas.

The topology within an area is hidden from the rest of the AS. Hiding this information enables a significant reduction in LSA traffic and reduces the computations needed to maintain the LSDB. Routing within the area is determined only by the topology of the area.

The three types of routers defined by OSPFv3 are as follows:

- Internal router (IR)—An internal router has all of its interfaces within the same area.
- Area border router (ABR)—An ABR has interfaces in multiple areas. It is responsible for exchanging summary advertisements with other ABRs.
- Autonomous system border router (ASBR)—An ASBR acts as a gateway between OSPFv3 and other routing protocols, or other autonomous systems.

Note
Configuring inter-area and external filters causes all of the neighbors in the area to be reset.

Backbone Area (Area 0.0.0.0)

Any *OSPFv3* network that contains more than one area is required to have an area configured as area 0.0.0.0, also called the backbone.

All areas in an AS must be connected to the backbone. When designing networks, you should start with area 0.0.0.0 and then expand into other areas.

Note
Area 0.0.0.0 exists by default and cannot be deleted or changed.

The backbone allows summary information to be exchanged between area border routers (ABRs). Every ABR hears the area summaries from all other ABRs. The ABR then forms a picture of the distance to all networks outside of its area by examining the collected advertisements and adding in the backbone distance to each advertising router.

When a *VLAN* is configured to run OSPFv3, you must configure the area for the VLAN.

If you want to configure the VLAN to be part of a different OSPFv3 area, use the following command:

```
configure ospfv3 vlan vlan-name area area-identifier
```

If this is the first instance of the OSPFv3 area being used, you must create the area first using the following command:

```
create ospfv3 area area_identifier
```

Stub Areas

OSPF allows certain areas to be configured as stub areas. A stub area is connected to only one other area. The area that connects to a stub area can be the backbone area. External route information is not distributed into stub areas. Stub areas are used to reduce memory consumption and computational requirements on OSPF routers.

Use the following command to configure an OSPF area as a stub area:

```
configure ospf area area-identifier stub [summary | nosummary] stub-default-cost cost
```

Not-So-Stubby-Areas

Not-so-stubby-areas (NSSAs) are similar to the existing *OSPF* stub area configuration option but have the following two additional capabilities:

- External routes originating from an ASBR connected to the NSSA can be advertised within the NSSA.
- External routes originating from the NSSA can be propagated to other areas, including the backbone area.

The CLI command to control the NSSA function is similar to the command used for configuring a stub area, as follows:

```
configure ospf area area-identifier nssa [summary | nosummary] stub-default-cost cost {translate}
```

The translate option determines whether type 7 LSAs are translated into type 5 LSAs. When configuring an OSPF area as an NSSA, translate should only be used on NSSA border routers, where translation is to be enforced. If translate is not used on any NSSA border router in a NSSA, one of the ABRs for that NSSA is elected to perform translation (as indicated in the NSSA specification). The option should not be used on NSSA internal routers. Doing so inhibits correct operation of the election algorithm.

Normal Area

A normal area is an area that is not:

- Area 0
- Stub area
- NSSA

Virtual links can be configured through normal areas. External routes can be distributed into normal areas.

Virtual Links

In the situation when a new area is introduced that does not have a direct physical attachment to the backbone, a virtual link is used.

A virtual link provides a logical path between the ABR of the disconnected area and the ABR of the normal area that connects to the backbone. A virtual link must be established between two ABRs that have a common area, with one ABR connected to the backbone. Figure 171 illustrates a virtual link.

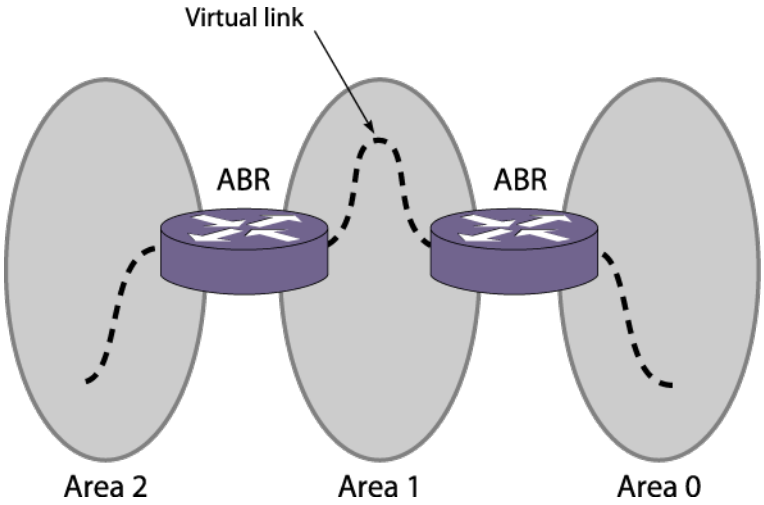


Figure 171: Virtual Link Using Area 1 as a Transit Area

Note
Virtual links cannot be configured through a stub or NSSA area.

Virtual links are also used to repair a discontiguous backbone area. For example, in Figure 172, if the connection between ABR 1 and the backbone fails, the connection using ABR 2 provides redundancy so that the discontiguous area can continue to communicate with the backbone using the virtual link.

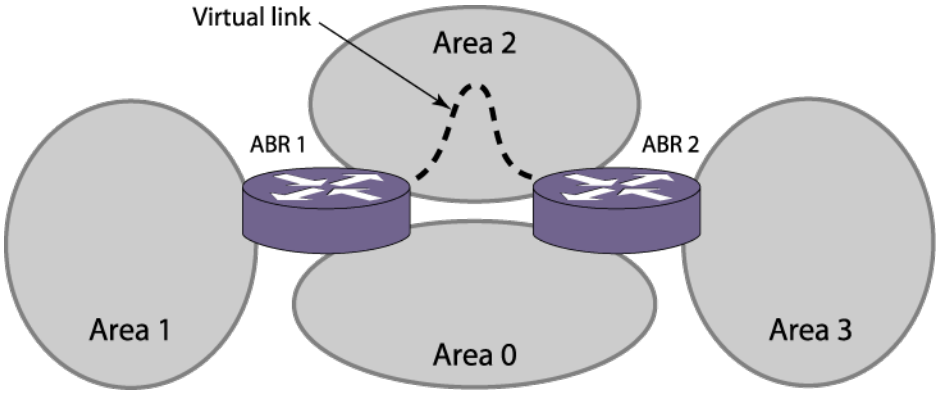


Figure 172: Virtual Link Providing Redundancy

Point-to-Point Support

You can manually configure the *OSPF* link type for a *VLAN*. Table 108 describes the link types.

Table 108: OSPF Link Types

Link Type	Number of Routers	Description
Auto	Varies	ExtremeXOS automatically determines the OSPF link type based on the interface type. This is the default setting.
Broadcast	Any	Routers must elect a designated router (DR) and a backup designated router (BDR) during synchronization. Ethernet is an example of a broadcast link.
Point-to-point	Up to 2	This type synchronizes faster than a broadcast link because routers do not elect a DR or BDR. It does not operate with more than two routers on the same VLAN. The Point-to-Point Protocol (PPP) is an example of a point-to-point link. An OSPF point-to-point link supports only zero to two OSPF routers and does not elect a designated router (DR) or backup designated router (BDR). If you have three or more routers on the VLAN, OSPF fails to synchronize if the neighbor is not configured.
Passive		A passive link does not send or receive OSPF packets.

Note

The number of routers in an OSPF point-to-point link is determined per VLAN, not per link.

All routers in the VLAN must have the same OSPF link type. If there is a mismatch, OSPF attempts to operate, but it may not be reliable.

Route Redistribution

More than one routing protocol can be enabled simultaneously on the switch.

Route redistribution allows the switch to exchange routes, including static routes, between the routing protocols. Figure 173 is an example of route redistribution between an OSPF AS and a RIP AS.

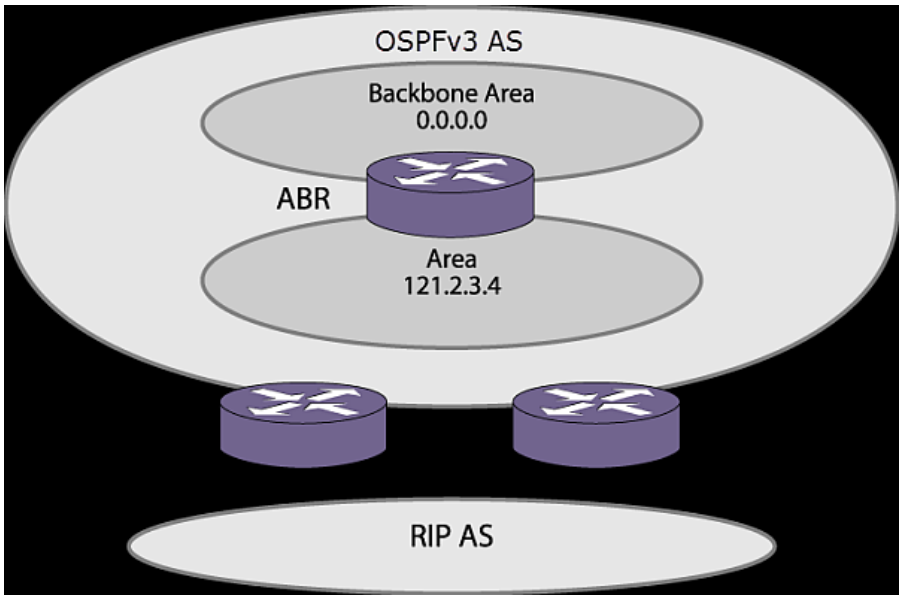


Figure 173: OSPF Route Redistribution

Import Policy

Prior to ExtremeXOS 15.7, routing protocol OSPFv2 applied routing policies with keyword “import-policy”, which can only be used to change the attributes of routes installed into the switch routing table. ExtremeXOS 15.7 provides the flexibility of using import policy to determine the routes to be added to or removed from the routing table.

To prevent a route being added to the routing table, the policy file must contain a matching rule with action “deny”. If there is no matching rule for a particular route, or the keyword “deny” is missing in the rule, the default action is “permit”, which means that route will be installed into the routing table. Refer to the following policy file example:

```
entry entry-one {
if {
nlri 11.22.0.0/16;
}
then {
cost 100;
}
}
entry entry-two {
if {
nlri 22.33.0.0/16;
}
then {
deny;
}
}
```

In the above policy example, entry-one is used to change the cost of any matching routes, and entry-two is used to remove those matching routes from the routing table.

Note
Only “Network Layer Reachability Information” (NLRI) and "route origin" can be used as matching criteria in policy rules; using "next_hop" as a matching criteria is not supported. Any other policy attribute is not recognized and is ignored.

Configuring Route Redistribution

Exporting routes from one protocol to another and in the reverse direction are discrete configuration functions. For example, to run IS-IS and *RIP* simultaneously, you must first configure both protocols and then verify the independent operation of each. Then you can configure the routes to export from IS-IS to RIP and the routes to export from RIP to IS-IS. Likewise, for any other combinations of protocols, you must separately configure each to export routes to the other.

This section describes how to inject routing information learned from other IP routing protocols into an IS-IS domain, thereby advertising their reachability in the IS-IS domain. These routes are included in the locally originated LSPs. For information on exporting routes learned in IS-IS to another routing protocol, see the chapter that describes the destination routing protocol.

When you export routes from another protocol to IS-IS, the metric and the type of metric are configured. You can also configure a policy to filter out unwanted external routes.

- To enable or disable the exporting of *BGP (Border Gateway Protocol)*, *OSPF*, RIP, static, and direct (interface) IPv4 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export {ipv4} route-type [policy | metric
  mvalue {metric-type [internal | external]}] {level[1| 2 | both-1-
  and-2]}
  ```

  ```
  disable isis area area_name export {ipv4} route-type
  ```

- To enable or disable the exporting of *OSPFv3*, *RIPng (Routing Information Protocol Next Generation)*, static, and direct (interface) IPv6 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export ipv6 route-type [policy | metric
  mvalue] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export ipv6 route-type
  ```

  The cost metric is inserted for all *OSPF*, RIP, static, and direct routes injected into IS-IS.

  The same cost and type values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route. The exported routes can also be filtered using policies.

- Verify the configuration using the command:

  ```
  show isis area [area_name | all]
  ```

Redistribute Routes into OSPF

To enable or disable the exporting of *BGP*, *RIP*, static, and direct (interface) routes to *OSPF*, use the following commands:

```
enable ospf export [bgp | direct | direct-inter-vr | e-bgp | i-bgp | rip
| static | isis | isis-level-1 | isis-level-1-external | isis-level-2
| isis-level-2-external | host-mobility] [cost cost type [ase-type-1 |
ase-type-2] {tag number} | policy-map]
```

```
disable ospf export [bgp | direct | direct-inter-vr | host-mobility |e-
bgp | i-bgp | rip | static | isis | isis-level-1| isis-level-1-external
| isis-level-2 | isis-level-2-external]
```

These commands enable or disable the exporting of RIP, static, and direct routes by way of LSA to other OSPF routers as AS-external type 1 or type 2 routes. The default setting is disabled.

The cost metric is inserted for all Border Gateway Protocol (BGP), RIP, static, and direct routes injected into OSPF. If the cost metric is set to 0, the cost is inserted from the route. For example, in the case of BGP export, the cost equals the multiple exit discriminator (MED) or the path length. The tag value is used only by special routing applications. Use 0 if you do not have specific requirements for using a tag. (The tag value in this instance has no relationship with IEEE 802.1Q *VLAN* tagging.)

The same cost, type, and tag values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route.

The exported routes can also be filtered using policies.

Note
For routes exported to OSPF via a policy file, any refresh applied on that policy may result in temporary withdrawal and then immediate readvertising of those routes.

Verify the configuration using the command `show ospf`.

OSPF Timers and Authentication

Configuring *OSPF* timers and authentication on a per-area basis is a shortcut to applying the timers and authentication to each *VLAN* in the area at the time of configuration. If you add more VLANs to the area, you must configure the timers and authentication for the new VLANs explicitly.

Use the command:

```
configure ospf vlan [vlan-name | all] timer retransmit-interval transit-
delay hello-interval dead-interval {wait-timer-interval}
```

OSPF Inter-VR Route Redistribution

Beginning with version 32.3, the OSPF Inter-VR route redistribution feature adds support for the redistribution of routes between multiple OSPF instances at the same time in different VRs. When inter-VR redistribution is enabled in the VR route table, OSPF uses Route Manager to install the routes.

Supported Platforms

All platforms.

CLI Commands

The following commands support this feature:

To configure an OSPF instance identifier, use `configure ospf instanceid instance-identifier`.

To disable an OSPF export, use `disable ospf export {vr} vr-name route-type`.

To enable an OSPF export, use `enable ospf export {vr} vr-name route-type [policy-map | cost cost type ase-type-1 | ase-type-2] {tag number}] {exclude-private}`.

To display OSPF Inter-VR export information, use `show ospf inter-vr-export` .

To display detailed OSPF Inter-VR export information, use `show ospf inter-vr-export {detail}`.

Configuring OSPF

Each switch that is configured to run *OSPF* must have a unique router ID.

We recommend that you manually set the router ID of the switches participating in OSPF, instead of having the switch automatically choose its router ID based on the highest interface IP address. Not performing this configuration in larger, dynamic environments could result in an older LSDB remaining in use.

Configuring OSPF Wait Interval

ExtremeXOS allows you to configure the *OSPF* wait interval, rather than using the router dead interval.

Caution
Do not configure OSPF timers unless you are comfortable exceeding OSPF specifications. Non-standard settings may not be reliable under all circumstances.

To specify the timer intervals, enter the following commands:

`configure ospf area area-identifier timer retransmit-interval transit-delay hello-interval dead-interval {wait-timer-interval}`

```
configure ospf virtual-link router-identifier area-identifier timer
retransmit-interval transit-delay hello-interval dead-interval

configure ospf vlan [vlan-name | all] timer retransmit-interval transit-
delay hello-interval dead-interval {wait-timer-interval}
```

OSPF Wait Interval Parameters

You can configure the following parameters:

- **Retransmit interval**—The length of time that the router waits before retransmitting an LSA that is not acknowledged. If you set an interval that is too short, unnecessary retransmissions result. The default value is 5 seconds.
- **Transit delay**—The length of time it takes to transmit an LSA packet over the interface. The transit delay must be greater than 0.
- **Hello interval**—The interval at which routers send hello packets. Shorter times allow routers to discover each other more quickly but also increase network traffic. The default value is 10 seconds.
- **Dead router wait interval (Dead Interval)**—The interval after which a neighboring router is declared down because hello packets are no longer received from the neighbor. This interval should be a multiple of the hello interval. The default value is 40 seconds.
- **Router wait interval (Wait Timer Interval)**—The interval between the interface coming up and the election of the DR and BDR. This interval should be greater than the hello interval. If this time is close to the hello interval, the network synchronizes very quickly but might not elect the correct DR or BDR. The default value is equal to the dead router wait interval.

Note
The *OSPF* standard specifies that wait times are equal to the dead router wait interval.

OSPF Configuration Example

Figure 174 is an example of an autonomous system using *OSPF* routers. The details of this network follow.

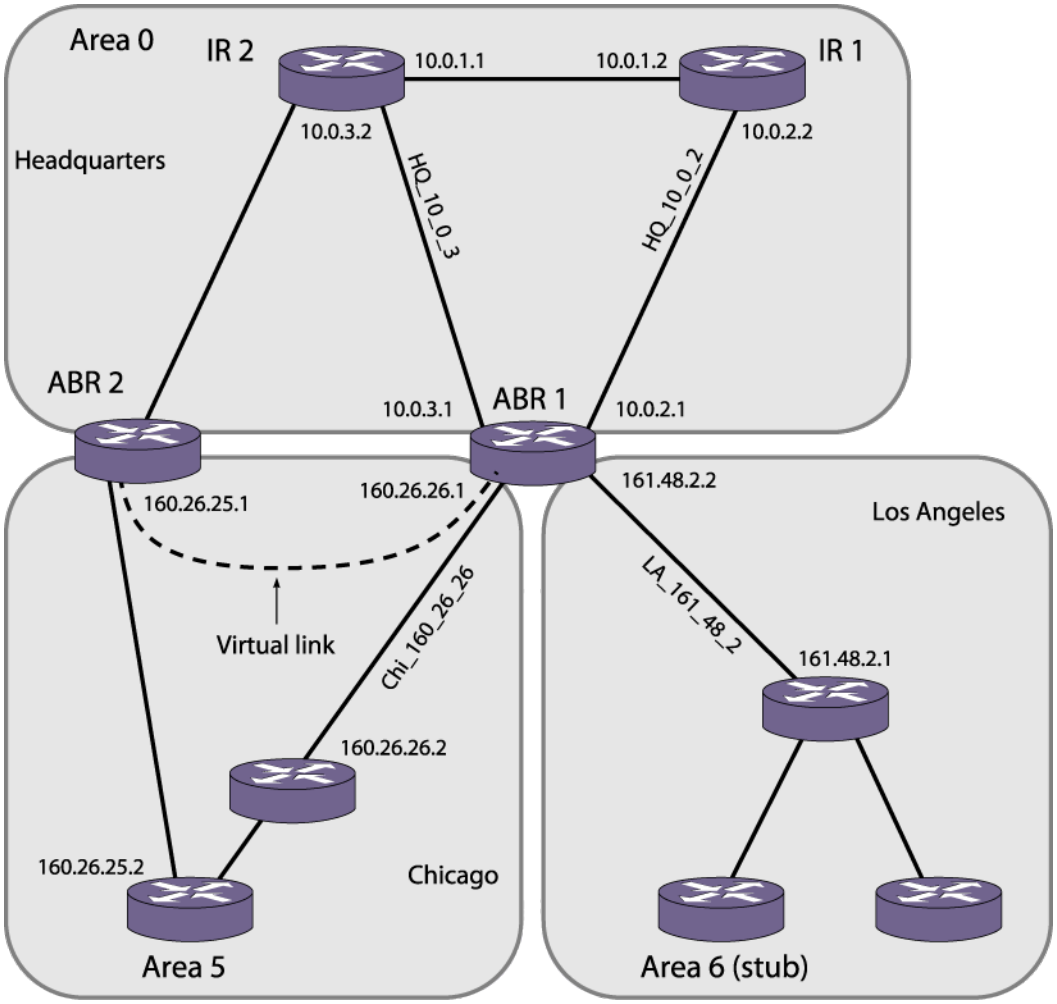


Figure 174: OSPF Configuration Example

Area 0 is the backbone area. It is located at the headquarters and has the following characteristics:

- Two internal routers (IR1 and IR2)
- Two area border routers (ABR 1 and ABR 2)
- Network number 10.0.x.x
- Two identified VLANs (HQ_10_0_2 and HQ_10_0_3)

Area 5 is connected to the backbone area by way of ABR 1 and ABR 2. It is located in Chicago and has the following characteristics:

- Network number 160.26.x.x
- One identified *VLAN* (Chi_160_26_26)
- Two internal routers

Area 6 is a stub area connected to the backbone by way of ABR 1. It is located in Los Angeles and has the following characteristics:

- Network number 161.48.x.x
- One identified VLAN (LA_161_48_2)
- Three internal routers
- Uses default routes for inter-area routing

Configuration for ABR 1 on page 1408 and Configuration for IR 1 on page 1408 provide two router configurations for the example in Figure 174 on page 1407.

Configuration for ABR 1

The router labeled ABR 1 has the following configuration:

```
create vlan HQ_10_0_2
create vlan HQ_10_0_3
create vlan LA_161_48_2
create vlan Chi_160_26_26
configure vlan HQ_10_0_2 ipaddress 10.0.2.1 255.255.255.0
configure vlan HQ_10_0_3 ipaddress 10.0.3.1 255.255.255.0
configure vlan LA_161_48_2 ipaddress 161.48.2.2 255.255.255.0
configure vlan Chi_160_26_26 ipaddress 160.26.26.1 255.255.255.0
create ospf area 0.0.0.5
create ospf area 0.0.0.6
enable ipforwarding
configure ospf area 0.0.0.6 stub nosummary stub-default-cost 10
configure ospf add vlan LA_161_48_2 area 0.0.0.6
configure ospf add vlan Chi_160_26_26 area 0.0.0.5
configure ospf add vlan HQ_10_0_2 area 0.0.0.0
configure ospf add vlan HQ_10_0_3 area 0.0.0.0
configure ospf vlan LA_161_48_2 priority 10
configure ospf vlan Chi_160_26_26 priority 10
configure ospf vlan HQ_10_0_2 priority 5
configure ospf vlan HQ_10_0_3 priority 5
enable ospf
```

Configuration for IR 1

The router labeled IR 1 has the following configuration:

```
configure vlan HQ_10_0_1 ipaddress 10.0.1.2 255.255.255.0
configure vlan HQ_10_0_2 ipaddress 10.0.2.2 255.255.255.0
enable ipforwarding
configure ospf add vlan all area 0.0.0.0
configure ospf area 0.0.0.0 priority 10
enable ospf
```

Note

In *OSPF* edge mode, the *VLAN* priority is “0” and cannot be set. (Refer to OSPF Edge Mode on page 1392.) When the license is upgraded to a Premier or Extreme Platform ONE license, the VLAN priority of "0" needs to be reset in order to participate in DR/BDR election.

Displaying OSPF Settings

You can use a number of commands to display settings for *OSPF*.

- To show global OSPF information, use the `show ospf` command with no options.
- To display information about one or all OSPF areas, enter:

  `show ospf area {`**`detail`** `|` *`area-identifier`*`}`

  The detail option displays information about all OSPF areas in a detail format.
- To display information about OSPF interfaces for an area, a *VLAN*, or for all interfaces enter:

  `show ospf interfaces {`**`vlan`** *`vlan-name`* `|` **`area`** *`area-identifier`* `|` **`enabled`**`}`

  The detail option displays information about all OSPF interfaces in a detail format.

  ExtremeXOS provides several filtering criteria for the `show ospf lsdb` command.

  You can specify multiple search criteria, and only those results matching all of the criteria are displayed. This allows you to control the displayed entries in large routing tables.
- To display the current link-state database, enter:

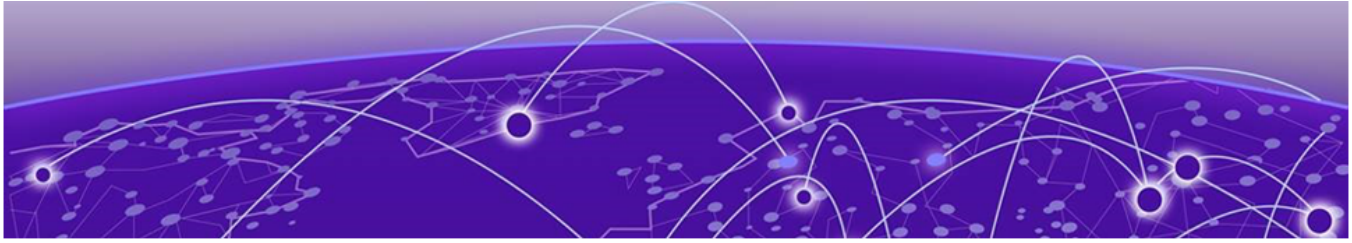
  `show ospf lsdb {`**`detail`** `|` **`stats`**`} {`**`area`** `[`*`area-identifier`* `|` **`all`**`]} {{`**`lstype`**`} [`*`lstype`* `|` **`all`**`]} {`**`lsid`** *`lsid-address`*`{`*`lsid-mask`*`}} {`**`routerid`** *`routerid-address`* `{`*`routerid-mask`*`}} {`**`interface`**`[[`*`ip-address`*`{`*`ip-mask`*`} | `*`ipNetmask`*`] | `**`vlan`** *`vlan-name`*`]}`

  The detail option displays all fields of matching LSAs in a multiline format. The summary option displays several important fields of matching LSAs, one line per LSA. The stats option displays the number of matching LSAs but not any of their contents. If not specified, the default is to display in the summary format.

  A common use of this command is to omit all optional parameters, resulting in the following shortened form:

  `show ospf lsdb`

  The shortened form displays LSAs from all areas and all types in a summary format.



OSPFv2 Auto-peering

OSPFv2 Auto-peering Introduction on page 1410
OSPFv2 Auto-peering Feature Description on page 1410
OSPFv2 Auto-peering Limitations on page 1411
OSPFv2 Auto-peering Supported Platforms on page 1411
Configuring OSFPv2 Auto-peering on page 1412

OSPFv2 Auto-peering Introduction

ExtremeXOS *OSPF (Open Shortest Path First)*v2 auto-peering is a network of cooperating interconnected devices that automatically create an OSPFv2 network for any topology, providing fully redundant, multipath routing. Node participation grows dynamically and is not bound to any well-known topology such as Clos or Leaf/Spine. OSPF is a link-state routing protocol that provides fast convergence and scalability. It is an interior gateway protocol (IGP) commonly used in large enterprise networks.

Auto-peering nodes build a secure network by running the OSPFv2 protocol over unnumbered point-to-point links to exchange database information that describes the topology of the autonomous system.

Any device connecting to an auto-peering device is an attachment point to the network. This network provides the underlay for services such as *VXLAN (Virtual Extensible LAN)*, policy, VRF, and Fabric Attach.

OSPFv2 Auto-peering Feature Description

After running the command `create` **`auto-peering ospf routerid`** *`ipaddress`*, the default OSPF configuration that is needed for auto-peering is created, and the AutoOSPFv2 capability is advertised out all ports using *LLDP (Link Layer Discovery Protocol)*. The default configuration includes the routerid, and vxlan-extensions and export of host-mobility routes. The following series of events occurs, setting up auto-peering:

1. When an OSPFv2 Auto-peering–capable neighbor is detected out a port, LLDP messages are sent.
2. LLDP-learned port is assigned to a new dynamically created VLAN, as an unnumbered interface.
3. LLDP learns of the OSPFv2 router ID.

4. LLDP sends the AutoOSPFv2 information to the remote neighbor and learns the remote neighbor's information.
5. An Auto-peering route is created to the learned remote router ID.
6. When OSPF learns of the auto-peering dynamic VLAN, it automatically enables a point-to-point link on that VLAN and places it in the backbone area.
7. The OSPF Auto-peering network is formed. OSPFv2 routes are propagated. The nexthop of the routes is the neighbor's router ID. The routes have the loose next hop flag set so the gateway can be resolved by the auto-peering route.
8. When OSPF Auto-peering is enabled, the OSPFv2 capability to carry VXLAN information is automatically enabled on each OSPFv2 peer.
9. VNI/LTEP pairs are passed to the OSPFv2 client which generates OSPFv2-opaque advertisements to any peers.
10. When an OSPFv2-opaque LSA is received containing a VNI/LTEP pair, the OSPFv2 client passes this to the client interface.

OSPFv2 Auto-peering Limitations

Note
BGP and OSPFv2 Auto-peering are mutually exclusive. Only one form of Auto-peering can exist at a time.

The following capabilities are not supported on OSPFv2 Auto-peering:

- Stacking
- Automatic LAG configuration with MLAG
- Automatic LAG configuration with Extended Edge Switching
- Static Lag attachments to AutoOSPF
- AutoOSPF links on Extended Edge Switching ports
- MPLS, VPLS, L2VPN, L3VPN
- PIM snooping, PIM-SM, PIM-DM, SSM mode
- OSPFv2, OSPFv3, ISIS, RIP, RIPNG per VRF
- Automatic MLAG configuration
- Multiple links between two OSPFv2 auto-peering enabled switches (instead LAG can be configured to avoid connectivity problem due to link failure).

Other limitations:

- One MLAG peer per node.
- Static router must be an external router per VRF.
- VLANs spanning multiple bridges, where each bridge has automatically configured LAG connections, must be VXLAN-based or replaced with MLAG.

OSPFv2 Auto-peering Supported Platforms

All ExtremeSwitching Universal platforms.

Configuring OSFPv2 Auto-peering

To create and enable OSPFv2 Auto-peering, use the following command:

```
create auto-peering ospf routerid ipaddress
```

To delete OSPFv2 Auto-peering, use the following command:

```
delete auto-peering
```

To view OSPFv2 and BGP auto-peering status, use the following command:

```
show auto-peering {bgp | ospf}.
```

OSPFv3

OSPFv3 Overview on page 1413

This chapter discusses the OSPFv3 (Open Shortest Path First version 3) protocol for distributing routing information between routers belonging to an autonomous system. This chapter provides an overview of the protocol's features and example configuration commands.

Note
OSPFv3 is available on platforms with an Advanced Edge (or higher) or Base (or higher) license. For information about OSPFv3 licensing, see the licensing guide.

OSPFv3 Overview

OSPF (Open Shortest Path First) is a link state protocol that distributes routing information between routers belonging to a single IP domain; the IP domain is also known as an *Autonomous System (AS)*.

In a link-state routing protocol, each router maintains a database describing the topology of the AS. Each participating router has an identical database for an area maintained from the perspective of that router.

From the *LSDB (Link State Database)*, each router constructs a tree of shortest paths, using itself as the root. The shortest path tree provides the route to each destination in the AS. When several equal-cost routes to a destination exist, traffic can be distributed among them. The cost of a route is described by a single metric.

OSPFv3 supports IPv6, and uses commands only slightly modified from that used to support IPv4. OSPFv3 has retained the use of the 4-byte, dotted decimal numbers for router IDs, LSA IDs, and area IDs.

OSPFv3 is an *IGP (Interior Gateway Protocol)*, as is the other common IGP for IPv6, *RIPng (Routing Information Protocol Next Generation)*. OSPFv3 and RIPng are compared in RIPng.

Note
Two types of OSPFv3 functionality are available and each has a different licensing requirement. One is the complete OSPFv3 functionality and the other is OSPFv3 Edge Mode, a subset of OSPFv3 that is described below. For specific information regarding OSPFv3 licensing, see the Switch Engine v33.6.1 Licensing Guide document.

OSPFv3 Edge Mode

OSPFv3 (Open Shortest Path First version 3) Edge Mode is a subset of OSPFv3 available on platforms with an Advanced Edge or Base license.

There are two restrictions on OSPFv3 Edge Mode:

- At most, four Active OSPFv3 *VLAN (Virtual LAN)* interfaces are permitted. There is no restriction on the number of Passive interfaces.
- The OSPFv3 Priority on VLANs is 0, and is not configurable. This prevents the system from acting as a DR or BDR.

BFD for OSPFv3

The BFD for *OSPFv3* feature gives OSPFv3 routing protocol the ability to utilize BFD's fast failure detection to monitor OSPFv3 neighbor adjacencies. CLI commands are provided to configure BFD protection for OSPFv3, so that as a registered BFD client, OSPFv3 can request BFD protection for interested OSPFv3 neighbors, and receive notifications about BFD session setup status and BFD session status updates (after establishing) from the BFD server. When BFD detects a communication failure between neighbors, it informs OSPFv3, which causes the OSPFv3 neighbor state be marked as "down." This allows OSPFv3 protocol to quickly begin network convergence and use alternate paths to the affected neighbor.

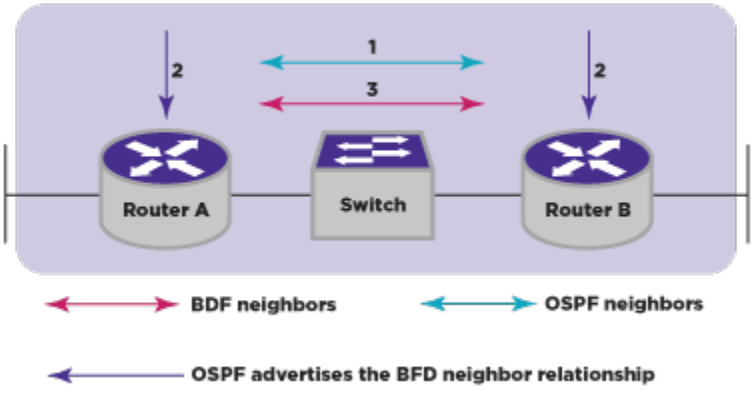


Figure 175: Basic Operational Flow of OSPFv3 BFD

Establishing a BFD Session for OSPFv3 Neighbor

1. OSPFv3 discovers a neighbor.
2. If BFD for OSPFv3 is configured, OSPFv3 on both routers sends a request to the local BFD server to initiate a BFD neighbor session with the OSPFv3 neighbor router.
3. The BFD neighbor session with the OSPFv3 neighbor router is established on both sides if BFD session limit is not reached.

If the BFD session limit is reached, the OSPFv3 neighbor will be marked as BFD session failed if synchronous request is used, or pending if asynchronous request is used, and the BFD server will send an asynchronous notification when the session registration passes later. (The asynchronous request is not available until the BFD client session create API is enhanced.)

Eliminating the OSPFv3 Neighbor Relationship by BFD Fault Detection

1. A failure occurs in the network.
2. The BFD neighbor session with the OSPFv3 neighbor router is removed because the BFD timer expired.
3. On both Router A and Router B, BFD notifies the local OSPFv3 process that the BFD neighbor is DOWN.
4. The local OSPFv3 process tears down the OSPFv3 neighbor relationship by marking neighbor state DOWN. (If an alternative path is available, the routers will immediately start converging on it.)

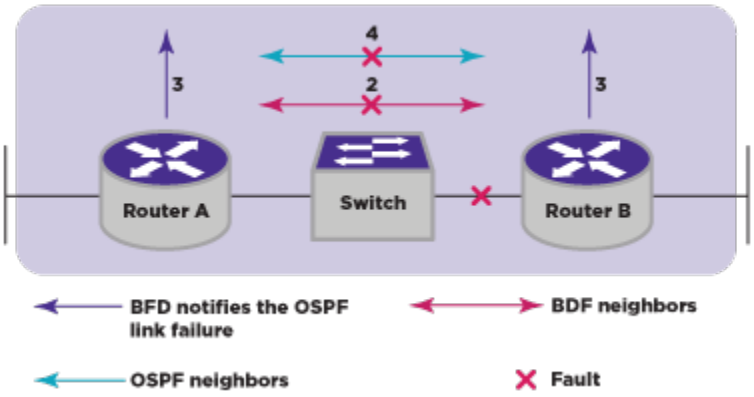


Figure 176: OSPFv3 Neighbor Relationship Eliminated by BFD Fault Detection

OSPFv3 will request the BFD server to delete the BFD session for OSPFv3 neighbor moved to DOWN state. When the link failure is later resolved, OSPFv3 needs to register the re-discovered neighbor to BFD again to initiate BFD session creation.

Removing BFD Protection for OSPFv3

1. BFD protection is removed from OSPFv3 interface on Router A, OSPFv3 process requests BFD server to delete all BFD sessions for neighbors learned on that interface.
2. BFD server will stop sending session status update to local OSPFv3 process on Router A.

3. Before actually deleting any sessions, BFD server on Router A will first notify Router B to mark those session status as "Admin Down," which will cause Router B to stop using BFD protection for those OSPFv3 neighbors.
4. When BFD protection is removed from OSPFv3 interface on Router B, BFD sessions can be deleted immediately since they are already in "Admin Down" state.

When BFD for OSPFv3 is configured on broadcast interface, the default behavior is to register only OSPFv3 neighbors in FULL state with the BFD server. Separate BFD sessions are created for each neighbor learned on the same interface. If multiple clients ask for the same neighbor on the same interface, then single BFD sessions are established between the peers.

OSPFv3 Neighbor State Determination

With active BFD protection, OSPFv3 combines the BFD session state with the associated interface admin and operational states to determine the OSPFv3 neighbor adjacency discovered on that OSPFv3 interface. Regarding OSPFv3 neighbor relationships, OSPFv3 reacts directly to BFD session state changes only in the following circumstances:

- If BFD is enabled on the interface, and
- If BFD for OSPFv3 is configured on the OSPFv3 interface, and
- If a BFD session has been established to the neighbor, and
- If the BFD session has passed the INIT_COMPLETE state then:

1. The OSPFv3 neighbor relationship will remain as FULL if the operational state of the BFD session is "UP" and the operational state of the associated *VLAN* interface is UP.
2. The OSPFv3 neighbor relationship will be considered as DOWN if the operational state of the BFD session is DOWN or the operational state of the associated VLAN interface is DOWN.

In all other cases, the BFD session state is not considered as part of the reported OSPFv3 neighbor state, and the OSPFv3 neighbor state reverts to the operational state of the OSPFv3 interface only. When the BFD session is in ADMIN_DOWN state, OSPFv3 ignores BFD events and OSPFv3 neighbor adjacency is not be affected by the BFD session state change.

Link State Database

Upon initialization, each router transmits a link state advertisement (LSA) on each of its interfaces. LSAs are collected by each router and entered into the LSDB of each router. After all LSAs are received, the router uses the link state database (LSDB) to calculate the best routes for use in the IP routing table. *OSPF* uses flooding to distribute LSAs between routers. Any change in routing information is sent to all of the routers in the network. All routers within an area have the exact same LSDB.

Table 109 describes LSA type numbers.

Table 109: LSA Type Numbers

Type Number	Description
1	Router LSA
2	Network LSA
3	Summary LSA
4	AS summary LSA
5	AS external LSA
7	NSSA external LSA
9	Link local—Opaque
10	Area scoping—Opaque
11	AS scoping—Opaque

Graceful OSPFv3 Restart

RFC 3623 describes a way for *OSPF* control functions to restart without disrupting traffic forwarding.

Without graceful restart, adjacent routers will assume that information previously received from the restarting router is stale and will not be used to forward traffic to that router. However, in many cases, two conditions exist that allow the router restarting *OSPFv3* to continue to forward traffic correctly. The first condition is that forwarding can continue while the control function is restarted. Most modern router system designs separate the forwarding function from the control function so that traffic can still be forwarded independent of the state of the OSPFv3 function. Routes learned through OSPFv3 remain in the routing table and packets continue to be forwarded. The second condition required for graceful restart is that the network remain stable during the restart period. If the network topology is not changing, the current routing table remains correct. Often, networks can remain stable during the time for restarting OSPFv3.

Restarting and Helper Mode

Routers involved with graceful restart fill one of two roles: the restarting router or the helper router.

With graceful restart, the router that is restarting sends out Grace-LSAs informing its neighbors that it is in graceful restart mode, how long the helper router should assist with the restart (the grace period), and why the restart occurred. If the neighboring routers are configured to help with the graceful restart (helper-mode), they will continue to advertise the restarting router as if it was fully adjacent. Traffic continues to be routed as though the restarting router is fully functional. If the network topology changes, the helper routers will stop advertising the restarting router. The helper router will continue in helper mode until the restarting router indicates successful termination of graceful restart, the Grace-LSAs expire, or the network topology changes. A router

can be configured for graceful restart, and for helper-mode separately. A router can be a helper when its neighbor restarts, and can in turn be helped by a neighbor if it restarts.

Planned and Unplanned Restarts

Two types of graceful restarts are defined: planned and unplanned.

A planned restart would occur if the software module for *OSPFv3* was upgraded, or if the router operator decided to restart the OSPFv3 control function for some reason. The router has advance warning, and is able to inform its neighbors in advance that OSPFv3 is restarting.

An unplanned restart would occur if there was some kind of system failure that caused a remote reboot or a crash of OSPFv3, or a failover occurs. As OSPFv3 restarts, it informs its neighbors that it is in the midst of an unplanned restart.

You can decide to configure a router to enter graceful restart for only planned restarts, for only unplanned restarts, or for both. Also, you can separately decide to configure a router to be a helper for only planned, only unplanned, or for both kinds of restarts.

Configuring Graceful OSPFv3 Restart

- Configure a router to perform graceful *OSPFv3* restart:
  `configure ospfv3 restart [none | planned | unplanned | both]`

  Since a router can act as a restart helper router to multiple neighbors, you will specify which neighbors to help.
- Configure a router interface to act as a graceful OSPFv3 restart helper:
  `configure ospfv3 [[vlan | tunnel] all | {vlan} vlan-name | {tunnel} tunnel-name | area area-identifier] restart-helper [none | planned | unplanned | both]`

  `configure ospfv3 virtual-link {routerid} router_identifier {area} area_identifier timer {retransmit-interval} retransmit_interval {transit-delay} transit_delay {hello-interval} hello_interval {dead-interval} dead_interval`
- The graceful restart period sent out to helper routers can be configured with the following command:
  `configure ospfv3 restart grace-period seconds`

  By default, a helper router will terminate graceful restart if received LSAs would affect the restarting router.

  This will occur when the restart-helper receives an LSA that will be flooded to the restarting router or when there is a changed LSA on the restarting router's retransmission list when graceful restart is initiated.
- Disable or enable helper router LSA check:
  `disable ospfv3 [[vlan | tunnel] all | vlanvlan-name | {tunnel} tunnel-name | area area-identifier] restart-helper-lsa-check`

  `enable ospfv3 [[vlan | tunnel] all | {vlan} vlan-name | {tunnel} tunnel-name | area area-identifier] restart-helper-lsa-check`

Areas

OSPFv3 allows parts of a network to be grouped together into areas.

The topology within an area is hidden from the rest of the AS. Hiding this information enables a significant reduction in LSA traffic and reduces the computations needed to maintain the LSDB. Routing within the area is determined only by the topology of the area.

The three types of routers defined by OSPFv3 are as follows:

- **Internal router (IR)**—An internal router has all of its interfaces within the same area.
- **Area border router (ABR)**—An ABR has interfaces in multiple areas. It is responsible for exchanging summary advertisements with other ABRs.
- **Autonomous system border router (ASBR)**—An ASBR acts as a gateway between OSPFv3 and other routing protocols, or other autonomous systems.

Note
Configuring inter-area and external filters causes all of the neighbors in the area to be reset.

Backbone Area (Area 0.0.0.0)

Any *OSPFv3* network that contains more than one area is required to have an area configured as area 0.0.0.0, also called the backbone.

All areas in an AS must be connected to the backbone. When designing networks, you should start with area 0.0.0.0 and then expand into other areas.

Note
Area 0.0.0.0 exists by default and cannot be deleted or changed.

The backbone allows summary information to be exchanged between area border routers (ABRs). Every ABR hears the area summaries from all other ABRs. The ABR then forms a picture of the distance to all networks outside of its area by examining the collected advertisements and adding in the backbone distance to each advertising router.

When a *VLAN* is configured to run OSPFv3, you must configure the area for the VLAN.

If you want to configure the VLAN to be part of a different OSPFv3 area, use the following command:

```
configure ospfv3 vlan vlan-name area area-identifier
```

If this is the first instance of the OSPFv3 area being used, you must create the area first using the following command:

```
create ospfv3 area area_identifier
```

Stub Areas

OSPF allows certain areas to be configured as stub areas. A stub area is connected to only one other area. The area that connects to a stub area can be the backbone area. External route information is not distributed into stub areas. Stub areas are used to reduce memory consumption and computational requirements on OSPF routers.

Use the following command to configure an OSPF area as a stub area:

```
configure ospf area area-identifier stub [summary | nosummary] stub-default-cost cost
```

Not-So-Stubby-Areas

Not-so-stubby-areas (NSSAs) are similar to the existing *OSPF* stub area configuration option but have the following two additional capabilities:

- External routes originating from an ASBR connected to the NSSA can be advertised within the NSSA.
- External routes originating from the NSSA can be propagated to other areas, including the backbone area.

The CLI command to control the NSSA function is similar to the command used for configuring a stub area, as follows:

```
configure ospf area area-identifier nssa [summary | nosummary] stub-default-cost cost {translate}
```

The translate option determines whether type 7 LSAs are translated into type 5 LSAs. When configuring an OSPF area as an NSSA, translate should only be used on NSSA border routers, where translation is to be enforced. If translate is not used on any NSSA border router in a NSSA, one of the ABRs for that NSSA is elected to perform translation (as indicated in the NSSA specification). The option should not be used on NSSA internal routers. Doing so inhibits correct operation of the election algorithm.

Normal Area

A normal area is an area that is not:

- Area 0
- Stub area
- NSSA

Virtual links can be configured through normal areas. External routes can be distributed into normal areas.

Virtual Links

In the situation when a new area is introduced that does not have a direct physical attachment to the backbone, a virtual link is used.

A virtual link provides a logical path between the ABR of the disconnected area and the ABR of the normal area that connects to the backbone. A virtual link must be

established between two ABRs that have a common area, with one ABR connected to the backbone. Figure 177 illustrates a virtual link.

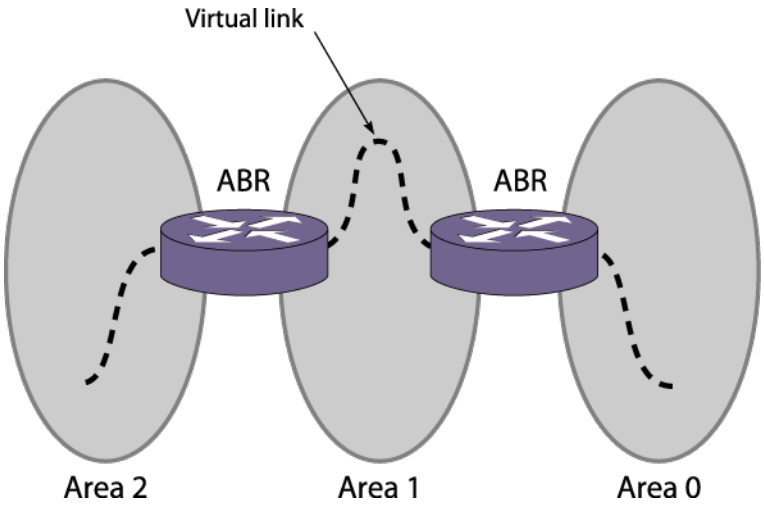


Figure 177: Virtual Link Using Area 1 as a Transit Area

Note
Virtual links cannot be configured through a stub or NSSA area.

Virtual links are also used to repair a discontiguous backbone area. For example, in Figure 178, if the connection between ABR 1 and the backbone fails, the connection using ABR 2 provides redundancy so that the discontiguous area can continue to communicate with the backbone using the virtual link.

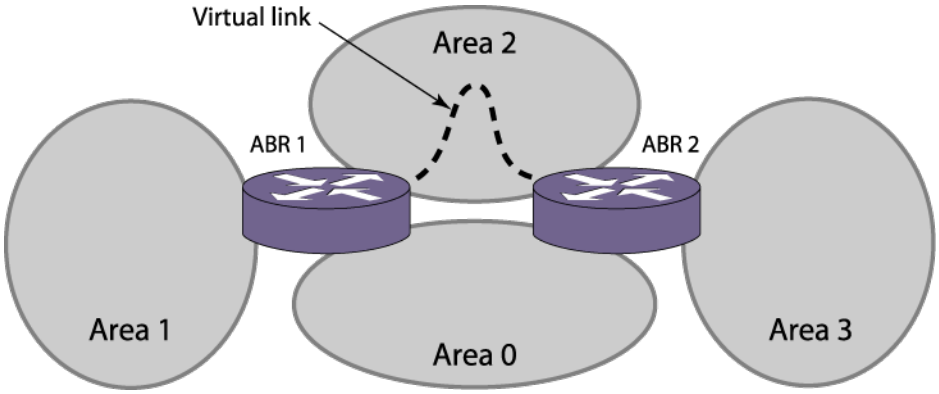


Figure 178: Virtual Link Providing Redundancy

Link-Type Support

You can manually configure the *OSPFv3* link type for a *VLAN*. The following table describes the link types.

Table 110: OSPFv3 Link Types

Link Type	Number of Routers	Description
Auto	Varies	ExtremeXOS automatically determines the OSPFv3 link type based on the interface type. This is the default setting.
Broadcast	Any	Routers must elect a designated router (DR) and a backup designated router (BDR) during synchronization. Ethernet is an example of a broadcast link.
Point-to-point	Up to 2	This type synchronizes faster than a broadcast link because routers do not elect a DR or BDR. It does not operate with more than two routers on the same VLAN. The Point-to-Point Protocol (PPP) is an example of a point-to-point link. An OSPFv3 point-to-point link supports only zero to two OSPFv3 routers and does not elect a designated router (DR) or backup designated router (BDR). If you have three or more routers on the VLAN, OSPF fails to synchronize if the neighbor is not configured.
Passive		A passive link does not send or receive OSPFv3 packets.

Note

The number of routers in an OSPFv3 point-to-point link is determined per VLAN, not per link.

All routers in the VLAN must have the same OSPFv3 link type. If there is a mismatch, OSPFv3 attempts to operate, but it may not be reliable.

Import Policy

Prior to ExtremeXOS 15.7, routing protocol OSPFv2 applied routing policies with keyword “import-policy”, which can only be used to change the attributes of routes installed into the switch routing table. ExtremeXOS 15.7 provides the flexibility of using import policy to determine the routes to be added to or removed from the routing table.

To prevent a route being added to the routing table, the policy file must contain a matching rule with action “deny”. If there is no matching rule for a particular route, or the keyword “deny” is missing in the rule, the default action is “permit”, which means that route will be installed into the routing table. Refer to the following policy file example:

```
entry entry-one {
if {
nlri 11.22.0.0/16;
}
then {
```

```
cost 100;
}
}
entry entry-two {
if {
nlri 22.33.0.0/16;
}
then {
deny;
}
}
```

In the above policy example, entry-one is used to change the cost of any matching routes, and entry-two is used to remove those matching routes from the routing table.

Note
Only “Network Layer Reachability Information” (NLRI) and "route origin" can be used as matching criteria in policy rules; using "next_hop" as a matching criteria is not supported. Any other policy attribute is not recognized and is ignored.

Route Redistribution

More than one routing protocol can be enabled simultaneously on the switch.

Route redistribution allows the switch to exchange routes, including static routes, between the routing protocols. Figure 179 is an example of route redistribution between an OSPF AS and a RIP AS.

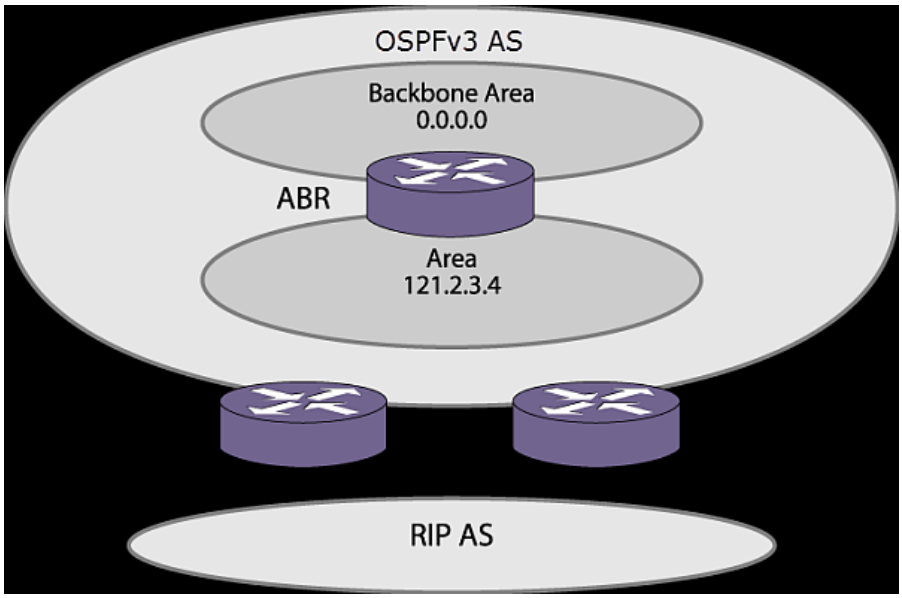


Figure 179: OSPF Route Redistribution

Configuring Route Redistribution

Exporting routes from one protocol to another and in the reverse direction are discrete configuration functions. For example, to run IS-IS and *RIP (Routing Information Protocol)* simultaneously, you must first configure both protocols and then verify the

independent operation of each. Then you can configure the routes to export from IS-IS to RIP and the routes to export from RIP to IS-IS. Likewise, for any other combinations of protocols, you must separately configure each to export routes to the other.

This section describes how to inject routing information learned from other IP routing protocols into an IS-IS domain, thereby advertising their reachability in the IS-IS domain. These routes are included in the locally originated LSPs. For information on exporting routes learned in IS-IS to another routing protocol, see the chapter that describes the destination routing protocol.

When you export routes from another protocol to IS-IS, the metric and the type of metric are configured. You can also configure a policy to filter out unwanted external routes.

- To enable or disable the exporting of *BGP (Border Gateway Protocol)*, *OSPF*, RIP, static, and direct (interface) IPv4 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export {ipv4} route-type [policy | metric mvalue {metric-type [internal | external]}] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export {ipv4} route-type
  ```

- To enable or disable the exporting of *OSPFv3*, *RIPng*, static, and direct (interface) IPv6 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export ipv6 route-type [policy | metric mvalue] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export ipv6 route-type
  ```

  The cost metric is inserted for all *OSPF*, RIP, static, and direct routes injected into IS-IS.

  The same cost and type values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route. The exported routes can also be filtered using policies.

- Verify the configuration using the command:

  ```
  show isis area [area_name | all]
  ```

Redistributing Routes into OSPFv3

To enable or disable the exporting of *RIPng*, static, and direct (interface) routes to *OSPFv3*, use the following command:

```
enable ospfv3 export [direct | ripng | static | isis | isis-level-1 | isis-level-1-external | isis-level-2 | isis-level-2-external| bgp | e-bgp | i-bgp |host-mobility] [cost cost type [ase-type-1 | ase-type-2] | policy_map]
```

These commands enable or disable the exporting of RIPng, static, and direct routes by way of LSA to other OSPFv3 routers as AS-external type 1 or type 2 routes. The default setting is disabled.

The cost metric is inserted for all RIPng, static, and direct routes injected into OSPFv3. If the cost metric is set to 0, the cost is inserted from the route. The tag value is used

only by special routing applications. Use 0 if you do not have specific requirements for using a tag. (The tag value in this instance has no relationship with IEEE 802.1Q *VLAN* tagging.)

The same cost, type, and tag values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route. The exported routes can also be filtered using policies.

Note
For routes exported to OSPFv3 via a policy file, any refresh applied on that policy may result in temporary withdrawal and then immediate readvertising of those routes.

Verify the configuration using the command:

```
show ospfv3
```

OSPFv3 Timers

Configuring *OSPFv3* timers on a per area basis is a shortcut to applying the timers to each *VLAN* in the area at the time of configuration.

If you add more VLANs to the area, you must configure the timers for the new VLANs explicitly. Use the command:

```
configure ospfv3 [vlan vlan-name | tunnel tunnel-name | [vlan |
tunnel] all] timer {retransmit-interval} retransmit-interval {transit-
delay} transit-delay {hello-interval} hello-interval {dead-interval}
dead-interval
```

Virtual Routing and Forwarding (VRF) for OSPFv3

ExtremeXOS supports multiple Virtual Routing and Forwarding (VRFs) in the same . Each VRF can have a separate instance of a routing protocol with its own routing table. *OSPFv3* can run in a Non-VPN VRF and support multiple instances of the protocol in different VRFs at the same time. This allows more scalable deployments of OSPFv3 as ExtremeXOS can have more VRFs than VRs, and OSPFv3 VRF instances belonging to the same VR run in a single ExtremeXOS process. Each VR still has its own process, so different VRs have very low impact on one another.

Limitations

- Only non-VPN VRFs are supported. OSPFv3 is blocked on VPN VRFs.
- *BGP* does not support VPN-IPv6 routes, so PE-CE support is not available.

OSPFv3 Authentication

There are two ways to perform authentication for OSPFv3: using IPsec and using Authentication Trailer. See the following for more information:

- IPsec Authentication on page 1426
- Authentication Trailer on page 1427

IPsec Authentication

IPsec is a framework for ensuring secure private communication over IP networks and is based on standards developed by the International Engineering Task Force (IETF). IPsec provides security services at the network layer of the Open Systems Interconnection (OSI) model by enabling a system to select required security protocols, determine the algorithms to use for the security services, and implement any cryptographic keys required to provide the requested services. IPsec can be used to protect one or more paths between a pair of hosts, between a pair of security gateways (such as switches), or between a security gateway and a host.

ExtremeXOS supports IPv6 Encapsulating Security Payload (ESP) in transport mode to provide authentication to OSPFv3 packets, and encryption is not supported. The transport mode Security Association (SA) is generally used between two hosts or routers/gateways when they are acting as hosts, and in OSPFv3 the routers assume the role of hosts as protocol packets are locally delivered. With ESP in transport mode, it will only provide authentication to OSPFv3 protocol packets excluding the IPv6 header, extension headers and options.

OSPFv3 exchanges both multicast and unicast packets. While running OSPFv3 over a broadcast interface, the authentication required is "one to many." Since IKE is based on the Diffie-Hellman key agreement protocol and works only for two communicating parties, it is not possible to use IKE for providing the required "one to many" authentication. *RFC4552* mandates the usage of Manual Keying with current IPsec implementation. In manual keying, SAs are statically installed on the routers and these static SAs are used to authenticate packets. it is not scalable and is practically infeasible to use different security associations for inbound and outbound traffic to provide the required "one to many" security. Therefore, *RFC4552* requires the implementations to use manually configured keys with the same SA parameters (for example, Security Parameter Index (SPI), keys, and so forth) for both inbound and outbound SAs.

Configuring OSPFv3 IPsec Authentication

To configure IPsec with a manual key to provide authentication for OSPFv3 interfaces, run the following command:

```
configure ospfv3 [{vlan} vlan-name | {tunnel} tunnel-name]
authentication [none |ipsec spi spi esp-auth-algorithm algorithm key
[key-string | encrypted encrypted-key-string]
```

To configure IPsec with a manual key to provide authentication on OSPFv3 virtual-links, run the following command:

```
configure ospfv3 virtual-link {routerid} router-identifier {area} area-
identifier authentication [none | ipsec spi spi esp-auth-algorithm
algorithm key [key-string | encrypted encrypted-key-string]
```

Authentication Trailer

Authentication Trailer for OSPFv3 (as described in RFC 7166) provides an alternative way to authenticate packets, as IPsec may not be suitable in some environments.

Authentication Trailer uses Keychain Manager to manage keys (see Keychain Manager Overview on page 1013). Keychain Manager provides OSPFv3 the key string and algorithm to use for authentication when a key becomes active, and it will notify OSPFv3 when a key expires. The authentication configuration is per interface or virtual interface, and the corresponding peers need to be configured with the same authentication keys. The maximum length of a key string that OSPFv3 can accommodate is 127 characters, which is the same as the maximum length of a key string currently allowed by Keychain Manager.

The cryptographic algorithms supported are HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

Note
OSPFv3 Authentication Trailer does not support the accept tolerance feature of Keychain Manager.

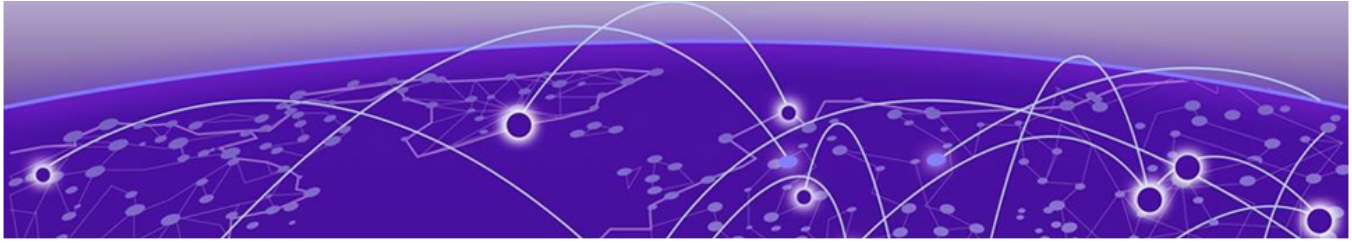
Configuring OSPFv3 Authentication Trailer

To configure Authentication Trailer to provide authentication for OSPFv3 interfaces see Configuring and Using Keychain Manager on page 1015, or run the following commands:

```
configure ospfv3 [{vlan} vlan-name | {tunnel} tunnel-name]
authentication [keychain keychain-name | none]
```

To configure Authentication trailer to provide authentication on OSPFv3 virtual-links, run the following command:

```
configure ospfv3 virtual-link {routerid} router-identifier {area} area-
identifier authentication [keychain keychain_name | none]
```



PoE

Extreme Networks PoE Devices on page 1428
Summary of PoE Features on page 1429
Power Delivery on page 1430
Configuring PoE on page 1437
Displaying PoE Settings and Statistics on page 1442

Power over Ethernet (PoE) is an effective method of supplying 48 VDC power to certain types of powered devices (PDs) through Category 5 and above twisted pair Ethernet cables.

PDs include wireless access points, IP telephones, laptop computers, web cameras, and other devices. With PoE, a single Ethernet cable supplies power and the data connection, reducing costs associated with separate power cabling and supply.

In a SummitStack, power is maintained across a failover on all PoE ports of non-primary nodes but is lost on all PoE ports of the failed primary node. Each ExtremeSwitching switch has its own PSU and the power budget for each ExtremeSwitching switch is determined by the internal/external PSUs connected to that ExtremeSwitching switch.

PoE+ supports higher power levels as defined by the IEEE 802.3at standard.

PoE++ supports higher power levels as defined by the IEEE 802.3bt standard.

Extreme Networks PoE Devices

Following is a list of the Extreme Networks devices that support PoE+ and PoE++ and the minimum required software:

PoE+

- 5420F-8W-16P-4XE—ExtremeXOS 31.3 and later.
- 5420F-24P-4XE—ExtremeXOS 31.3 and later.
- 5420F-16MW-32P-4XE—ExtremeXOS 31.3 and later.
- 5420F-16W-32P-4XE—ExtremeXOS 31.3 and later.
- 5420F-48P-4XE—ExtremeXOS 31.3 and later.
- 5420F-48P-4XL—ExtremeXOS 31.3 and later.
- 5420M-16MW-32P-4YE—ExtremeXOS 31.3 and later.
- 5320-48P-8XE—Switch Engine 31.6 and later.

- 5320-24P-8XE—Switch Engine 31.6 and later.
- 5320-16P-4XE—Switch Engine 31.7 and later.
- 5320-16P-4XE-DC—Switch Engine 31.7 and later.
- 5320-16P-2MXT-2X—Switch Engine 33.4.1 and later.
- 4220-12P-4X—Switch Engine 32.7.1 and later.
- 4220-24P-4X—Switch Engine 32.7.1 and later.
- 4220-48P-4X—Switch Engine 32.7.1 and later.
- 4220-4MW-8P-4X—Switch Engine 32.7.1 and later.
- 4220-4MW-20P-4X—Switch Engine 32.7.1 and later.
- 4220-8MW-40P-4X—Switch Engine 32.7.1 and later.

PoE++

- 5520-24W—ExtremeXOS 31.1 and later.
- 5520-48W—ExtremeXOS 31.1 and later.
- 5520-12MW-36W—ExtremeXOS 31.1 and later.
- 5420F-8W-16P-4XE—ExtremeXOS 31.3 and later.
- 5420F-16MW-32P-4XE—ExtremeXOS 31.3 and later.
- 5420F-16W-32P-4XE—ExtremeXOS 31.3 and later.
- 5420M-24W-4YE—ExtremeXOS 31.3 and later.
- 5420M-16MW-32P-4YE—ExtremeXOS 31.3 and later.
- 5420M-48W-4YE—ExtremeXOS 31.3 and later.
- 5720-24MW—Switch Engine 32.1 and later.
- 5720-24MXW—Switch Engine 32.1 and later.
- 5720-48MW—Switch Engine 32.1 and later.
- 5720-48MXW—Switch Engine 32.1 and later.
- 4120-24MW-4Y—Switch Engine 32.7 1 and later.
- 4120-48MW-4Y—Switch Engine 32.7 1 and later.
- 4220-4MW-8P-4X—Switch Engine 32.7 1 and later.
- 4220-4MW-20P-4X—Switch Engine 32.7 1 and later.
- 4220-8MW-40P-4X—Switch Engine 32.7 1 and later.
- 5420M-24W-24S-4YE— version 33.5.1 and later.

Summary of PoE Features

The ExtremeXOS implementation of *PoE (Power over Ethernet)* supports the following features:

- Configuration and control of the power distribution for PoE at the system, slot, and port levels
- Real-time discovery and classification of IEEE 802.3af-compliant PDs and many legacy devices
- Support for IEEE 802.3at-compliant PDs on PoE+ devices

- Support for IEEE 802.3bt compliant PDs on PoE++ devices
- Monitor and control of port PoE fault conditions including exceeding configured class limits and power limits and short-circuit detection
- Support for configuring and monitoring PoE status at the system, slot, and port levels
- Management of an over-subscribed power budget
- Port LED control for indicating the link state
- Support for hitless failover in a SummitStack with a master and backup node.
- (Fast PoE) Ability to provide PoE power when the switch is powered on without waiting for boot up based on last saved PoE state on all Universal (per port) platforms).
- (Perpetual PoE) Preserve PoE power delivery to devices during reboot on all Universal platforms.

Power Delivery

This section describes how the system provides power to the PDs. ExtremeXOS always configures the PoE controller to use Dynamic Power Management mode.

Enabling PoE to the Switch

PoE is enabled by default. For details about changing the configuration, see Configuring PoE on page 1437.

Power Reserve Budget

Power supply equipment (PSEs) can be categorized as either Type 1, Type 2, Type 3, or Type 4 PSEs . Type 3 PSEs support powered devices (PDs) up to class 6. Type 4 PSEs support PDs up to class 8.

4120 Series Switches

Switch Model	With Internal Fixed PSU	With Internal Fixed PSU + 920W PSU	With Internal Fixed PSU + 1200W PSU	With Internal Fixed PSU + 2000W PSU
4120-24MW-4Y (100-120VAC)	680W	1440W	1380W	Not Supported
4120-48MW-4Y (100-120VAC)	640W	Not Supported	1340W	1230W
4120-24MW-4Y (200-240VAC)	680W	1440W	1540W	Not Supported
4120-48MW-4Y (200-240VAC)	980W	Not Supported	1780W	2180W

4220 Series Switches

Switch Model	PoE Budget
4220-12P-4X	124W
4220-24P-4X	195W
4220-48P-4X	370W
4220-4MW-8P-4X	240W
4220-4MW-20P-4X	370W
4220-8MW-40P-4X	740W

5320-16P Series Switches

Switch Model	PoE Budget
5320-16P-2MXT-2X	180W
5320-16P-4XE	185W
5320-16P-4XE-DC	185W

5320-24P Series Switches

5320-24P series switches have the following power reserve budgets:

Table 111: 5320-24P-8XE PoE Power Budget

Single Internal Fixed PSU
370W

5320-48P Series Switches

5320-48P series switches have the following power reserve budgets:

Table 112: 5320-48P-8XE PoE Power Budget

Single Internal Fixed PSU
740W

5420F Series Switches

5420F series switches have the following power reserve budgets:

Table 113: 5420F-24P-4XE PoE Power Budget

Single Internal Fixed PSU	Recommended Secondary PSU
380W	720W

Table 114: 5420F-48P-4XE PoE Power Budget

Single Internal Fixed PSU	Recommended Secondary PSU
740W	1480W

Table 115: 5420F-48P-4XL PoE Power Budget

Single Internal Fixed PSU	Recommended Secondary PSU
740W	1480W

Table 116: 5420F-8W-16P-4XE PoE Power Budget

Single Internal Fixed PSU	Recommended Secondary PSU
480W	960W

Table 117: 5420F-16W-32P-4XE PoE Power Budget

Single Internal Fixed PSU	Recommended Secondary PSU
872W*/960W**	1437W*/1874W**

Table 118: 5420F-16MW-32P-4XE PoE Power Budget

Single Internal Fixed PSU	Recommended Secondary PSU
860W*/960W**	1425W*/1862W**

Note
* @ 100-120 VAC

Note
** @ 200-240VAC

5420M Series Switches

5420M series switches have the following power reserve budgets:

Table 119: 5420M-24W-4YE and 5420M-24W-24S-4YE PoE Power Budget

1 x 920W	2 x 920W
720W	1440W

Table 120: 5420M-48W-4YE PoE Power Budget

1 x 920W	2 x 920W	1 x 2000W @ 100-120V AC	1 x 2000W @ 200-240V AC	1 x 920W & 1 x 2000W @100-120 VAC	1 x 920W & 1 x 2000W @ 200-240V AC	2 x 2000W @ 100-120V AC	2 x 2000W @ 200-240V AC
804W	1558W	884W	1440W	1212W	2076W	1704W	2880W

Table 121: 5420M-16MW-32P-4YE PoE Power Budget

1 x 920W	2 x 920W	1 x 2000W @ 100-120V AC	1 x 2000W @ 200-240V AC	1 x 920W & 1 x 2000W @100-120 VAC	1 x 920W & 1 x 2000W @ 200-240V AC	2 x 2000W @ 100-120V AC	2 x 2000W @ 200-240V AC
787W	1541W	867W	1440W	1195W	2059W	1687W	2400W

5520 Series Switches

5520 series switches have the following power reserve budgets:

Table 122: 5520-24W—System Power: 400W

PSU1	PSU2	30W Ports	60W Ports	90W Ports	PoE Max
2,000W high line	2,000W high line	24	24	24	2,160W
20,00W low line	2,000W low line	24	24	20	1,800W
1,100W	1,100W	24	24	20	1,800W
715W	715W	24	17	11	1,020W
2,000W high line	0W	24	24	17	1,530W
2,000W low line	0W	23	11	7	690W

Table 122: 5520-24W—System Power: 400W (continued)

PSU1	PSU2	30W Ports	60W Ports	90W Ports	PoE Max
1,100W	0W	23	11	7	690W
715W	0W	10	5	3	300W

Table 123: 5520-48W—System Power: 400W

PSU1	PSU2	30W Ports	60W Ports	90W Ports	PoE Max
2,000W high line	2,000W high line	48	48	40	3,600W
2,000W low line	2,000W low line	48	30	20	1,800W
1,100W	1,100W	48	30	20	1,800W
715W	715W	34	17	11	1,020W
2,000W high line	0W	48	26	17	1,560W
2,000W low line	0W	23	11	7	690W
1,100W	0W	23	11	7	690W
715W	0W	10	5	3	300W

Table 124: 5520-12MW-36W—System Power: 400W

PSU1	PSU2	30W Ports	60W Ports	90W Ports	PoE Max
2,000W high line	2,000W high line	48	48	40	3,600W
2,000W low line	2,000W low line	48	30	20	1,800W
1,100W	1,100W	48	30	20	1,800W
715W	715W	34	17	11	1,020W
2,000W high line	0W	48	26	17	1,560W
2,000W low line	0W	23	11	7	690W
1,100W	0W	23	11	7	690W
715W	0W	10	5	3	300W

5720 Series Switches

series switches have the following power reserve budgets:

Table 125: 5720 PoE Power Budget

Switch Model	1 x 715W	2 x 715W	1 x 1100W	2 x 1100W	1 x 2000W @110-132 VAC	2 x 2000W @110-132 VAC	1 x 2000W @220-264VAC	2 x 2000W @220-264VAC
5720-24MW	450W	1093W	835W	1825W	835W	1825W	1735W	2160W
5720-24MXW	450W	1093W	835W	1825W	835W	1825W	1735W	2160W
5720-48MW	450W	1093W	835W	1825W	835W	1825W	1735W	3535W
5720-48MXW	450W	1093W	835W	1825W	835W	1825W	1735W	3535W

Note
It's recommended that primary and secondary PSUs be of the same type to support optimal PoE operation.

Guard Band

To reduce the chances of ports fluctuating between powered and non-powered states, newly inserted PDs are not powered when the actual delivered power for the switch is within a preset value below the configured inline power budget for that slot or switch.

This band is called the guard band and the value used is 20 W for all series switches. However, actual aggregate power can be delivered up to the configured inline power budget for the slot or switch (for example, when delivered power from ports increases or when the configured inline power budget for the slot is reduced).

If total consumed power is within the guard band, new ports are not powered up.

PD Disconnect Precedence

After a PD is discovered and powered on an ExtremeSwitching switch, the actual power drain is continuously measured.

If the usage for power by PDs is within the guard band, the system begins denying power to PDs.

You can configure the switch to handle a request for power that exceeds the power budget situation in one of two ways, called the disconnect precedence:

- Disconnect PDs according to the configured *PoE* port priority for each PD.
- Deny power to the next PD requesting power, regardless of that port's PoE priority. The default value is deny-port.

Port Priority Enabled

- When a PD is detected on a specific port, the PoE controller allows power on the port if available power form the power source is greater than the maximum power threshold configured on the port and the total allocated power is below the guard band.
- When available power is less than the maximum power threshold configured on the port and port priority is less than the port priority set on all powered ports, the PoE controller does not allow power on the port.
- When available power is less than the maximum power threshold configured on the port and the port priority is greater than the priority of currently powered ports, the PoE controller disconnects the two priority ports(s) and allows the port to power up.
- When the total consumed power exceeds the configured power limit of the power source, the PoE controller disconnects the low priority ports(s) and allows the port to power up.

Port Priority Disabled

- When a PD is detected on a specific port, the PoE controller allows power on the port if available power from the power source is greater than the maximum power threshold configured on the port and the total allocated power is below the guard band. Note that "maximum power threshold" is the same as "operator-limit".
- When available power is less than the maximum power threshold configured on the port, the PoE controller does not allow power on this port.
- When the total consumed power exceeds the configured power limit of the power source, the PoE controller starts powering down ports beginning with the highest port number.

Port Disconnect or Fault

On all *PoE* devices, when a port enters a fault state if you set the operator limit lower than the amount requested by the PD, the system removes power from that port. The power removed is, again, available only to other ports on the same slot or stand-alone switch. The port stays in the fault state until you disable that port, disconnect the attached PD, or reconfigure the operator limit to be high enough to satisfy the PD requirements.

When a port is disconnected or otherwise moves into a fault state, *SNMP (Simple Network Management Protocol)* generates an event (after you configure SNMP and a log message is created).

PoE Usage Threshold

The system generates an *SNMP* event when any slot or stand-alone switch has consumed a specified percentage of that slot's reserved power budget or of the entire power for the stand-alone switch.

The default value is 70%; you can configure this threshold to generate events from 1% to 99% consumption of the reserved power budget. You can also configure the system

to log an EMS message when the usage threshold is crossed (for more information on EMS, see Using the Event Management System/Logging on page 1750).

Legacy Devices

ExtremeXOS software allows the use of non-standard PDs with the switch. These are PDs that do not comply with the IEEE 802.3af standard.

The system detects non-standard PDs using a capacitance measurement. You must enable the switch to detect legacy devices; the default value is disabled.

Detecting a PD through capacitance is used only if the following two conditions are both met:

- Legacy PD detection is enabled.
- The system unsuccessfully attempted to discover the PD using the standard resistance measurement method.

PoE Operator Limits

You can set the power limit that a PD can draw on the specified ports. For *PoE*, the range is 3,000 to 16800mW and the default value is 15,400 mW. For PoE+, the range is 3,000 to 32,000 mW and the default value is 30,000 mW. For PoE++ Type 3, the range is 3,000 mW to 64,000 mW and the default value is 60,000 mW. For PoE++ Type 4, the range is 3,000 mW to 98,000 mW and the default value is 90,000 mW.

If the measured power for a specified port exceeds the port's operator limit, the power is withdrawn from that port and the port moves into a fault state.

If you attempt to set an operator limit outside the accepted range, the system returns an error message.

With a class-based operator-limit type, if a class4 PD is attached, then the operator-limit set will be ignored and the switch will not deliver more than 30W to the PD.

When class-based operator-limit is configured on a port, "Max Allowed Power" will show the maximum power for the class of the attached PD. Without the class-based option, "Max Allowed Power" will show the configured or default value of operator-limit.

Configuring PoE

PoE supports a full set of configuration and monitoring commands that allows you to configure, manage, and display PoE settings at the system, slot, and port level. To enable inline power, or PoE, you must have a powered switch.

To configure inline power, or PoE, you must accomplish the following tasks:

1. Enable inline power to the system, slot, and/or port.
2. On ExtremeSwitching all Universal switches, configure the disconnect precedence for the PDs in the case of excessive power demands.
3. Configure the threshold for initiating system alarms on power usage.

Additionally, you can configure the switch to use legacy PDs, apply specified PoE limits to ports, apply labels to PoE ports, and configure the switch to allow you to reset a PD without losing its power allocation.

For complete information on using the CLI commands, see the Switch Engine v33.6.1 Command References .

enable inline-power

```
enable inline-power [{fast {ports [port_list | all]}} | perpetual]
```

Description

Enables *PoE* and perpetual PoE to all ports; or fast PoE to all ports, or selected ports, for some platforms.

Syntax Description

fast	Deliver PoE power to devices at the time of switch power on without waiting for boot up based on last saved PoE state. The default is disabled.
ports	For fast PoE, specifies selecting ports. 4120, 4220, ExtremeSwitching 5320, 5420, 5520 and 5720 series switches only.
port_list	For fast PoE, specifies the port list separated by a comma or -. 4120, 4220, ExtremeSwitching 5320, 5420, 5520 and 5720 series switches only.
all	For fast PoE, specifies selecting all ports. 4120, 4220, ExtremeSwitching 5320, 5420, 5520 and 5720 series switches only.
perpetual	Preserves PoE power delivery to devices during reboot. Perpetual PoE is a switch-wide setting. The default is disabled.

Default

By default:

- PoE is enabled.
- Fast PoE is disabled.
- Perpetual PoE is disabled.

Usage Guidelines

You can control whether inline power is provided to the system by using the `disable inline-power` command and the `enable inline-power` command. By default, inline power provided to all ports is enabled. Additionally, you can opt to deliver PoE power to devices at the time of switch power on without waiting for boot up (fast PoE) based on last saved PoE state. Per-port fast PoE is available on certain platforms. You can also elect to preserve PoE power delivery to devices during reboot (perpetual PoE). The default for both PoE options is disabled.

Enabling inline power starts the PoE detection process used to discover, classify, and power remote PDs.

Disabling inline power using the `disable inline-power` command does not affect the data traffic traversing the port. And, disabling the port using the `disable port` command does not affect the inline power supplied to the port.

Note
Inline power cannot be delivered to connected PDs unless the switch is powered on.

Example

The following command enables inline power currently provided to all ports:

```
# enable inline-power
```

The following example turns on perpetual PoE for the switch:

```
# enable inline-power perpetual
```

The following example turns on fast PoE for ports 1,2, and 5:

```
# enable inline-power fast ports 1,2,5
```

History

This command was first available in ExtremeXOS 11.1.

The fast and perpetual PoE options were added in ExtremeXOS 30.3.

Per-port fast PoE was added for ExtremeXOS 31.1.

Platform Availability

This command is available on the PoE devices listed in Extreme Networks PoE Devices.

The fast and perpetual options are only available on the 4120, 4220, ExtremeSwitching 5320, 5420, 5520 and 5720 (per port) series switches.

Setting the Disconnect Precedence

All ExtremeSwitching Unversal *PoE* switches.

Note
The switch generates an *SNMP* event if a PD goes offline, and the port's state moves from Power to Searching. You must configure SNMP to generate this event.

When the actual power used by the PDs on a switch or slot exceeds the power budgeted for that switch or slot, the switch refuses power to PDs. There are two methods used by the switch to refuse power to PDs, and whichever method is in place applies to all PoE slots in the switch. This is called the disconnect precedence method, and you configure one method for the entire switch.

The available disconnect precedence methods are:

- Deny port
- Lowest priority

The default value is deny port. Using this method, the switch simply denies power to the next PD requesting power from the slot, regardless of that port's PoE priority or port number.

Using the lowest priority method of disconnect precedence, the switch disconnects the PDs connected to ports configured with lower PoE priorities.

When several ports have the same PoE priority, the lower port numbers have higher PoE priorities. That is, the switch withdraws power (or disconnects) those ports with the highest port number(s).

The system keeps dropping ports, using the algorithm you selected with the disconnect ports command, until the measured inline power for the slot is lower than the reserved inline power.

- To Configure the disconnect precedence for the switch, use the following command:

  ```
  configure inline-power disconnect-precedence [deny-port | lowest-priority]
  ```

- To return the disconnect precedence to the default value of deny port, use the following command:

  ```
  unconfigure inline-power disconnect-precedence
  ```

- To display the currently configured disconnect precedence, use the following command:

  ```
  show inline-power
  ```

Configuring the Usage Threshold

The system generates an *SNMP* event after a preset percentage of the reserved power for any slot or total power for a stand-alone switch is actually used by a connected PD. This preset percentage is called the usage threshold and is the percentage of the measured power to the budgeted power for each slot or total power for a stand-alone switch.

The default value for this usage threshold is 70%. You can configure the usage threshold to be any integer between 1% and 99%.

- To configure the threshold percentage of budgeted power used on a slot or the total power on a stand-alone switch that causes the system to generate an SNMP event and EMS message, use the following command:

  ```
  configure inline-power usage-threshold threshold
  ```

- To reset the threshold that causes the system to generate an SNMP event and EMS message per slot to 70% for measured power compared to budgeted power, use the following command:

  ```
  unconfigure inline-power usage-threshold
  ```

- To display the currently configured usage threshold, use the following command:

  ```
  show inline-power
  ```

Configuring the Switch to Detect Legacy PDs

The *PoE* device can detect non-standard, legacy PDs, which do not conform to the IEEE 802.3af standard, using a capacitance measurement. However, you must specifically enable the switch to detect these non-standard PDs; the default value for this detection method is disabled.

This configuration applies to the entire switch; you cannot configure the detection method per slot.

The switch detects PDs through capacitance only if both of the following conditions are met:

- The legacy detection method is enabled.
- The switch unsuccessfully attempted to discover the PD using the standard resistance measurement method.

- To configure a switch to detect legacy, non-standard PDs for a specified slot, use the following command:

  ```
  configure inline-power detection [802.3af-only | legacy-and-802.3af [4-point | 2-point] | bypass] ports port_list
  ```

- To configure the switch to detect legacy PDs on a switch, use the following command:

  ```
  configure inline-power detection [802.3af-only | legacy-and-802.3af [4-point | 2-point] | bypass] ports port_list
  ```

Configuring the Operator Limit

You can configure the maximum amount of power that the specified port can deliver to the connected PD, in milliwatts (mW). For *PoE*, the default value is 15400 mW, and the range is 3000 to 16800 mW. For PoE+, the range is 3,000 to 32,000 mW and the default value is 30,000 mW. For PoE++ Type 3, the range is 3,000 mW to 64,000 mW and the default value is 60,000 mW. For PoE++ Type 4, the range is 3,000 mW to 98,000 mW and the default value is 90,000 mW.

If the operator limit for a specified port is less than the power drawn by the legacy PD, the legacy PD is denied power.

- To set the operator limit on specified ports, which limits how much power a PD can draw from that port, use the following command:

  ```
  configure inline-power operator-limit milliwatts ports [all | port_list]
  ```

- To reset the operator limit to the default value of 15.4 W for PoE, 30 W for PoE+, or 60 W for PoE++, use the following command:

  ```
  unconfigure inline-power operator-limit ports [all |port_list]
  ```

- To display the current operator limit on each port, use the following command:

  `show inline-power configuration ports {port_list}`

  Note

  With a class-based operator-limit type, if a class4 PD is attached, then the operator-limit set will be ignored and the switch will not deliver more than 30W to the PD.

  When class-based operator-limit is configured on a port, "Max Allowed Power" will show the maximum power for the class of the attached PD. Without the class-based option, "Max Allowed Power" will show the configured or default value of operator-limit.

Configuring PoE Port Labels

You can assign labels to a single or group of *PoE* ports using a string of up to 15 characters.

- To assign a label to PoE ports, use the following command:

  `configure inline-power label [string | none] ports port_list`

  To rename a port or to return it to a blank label, reissue the command.
- To display the PoE port labels, use the following command:

  `show inline-power configuration ports {port_list}`

Power Cycling Connected Powered Devices

You can power cycle a connected PD without losing the power allocated to its port using the following command:

`reset inline-power ports port_list`

Displaying PoE Settings and Statistics

You can display the *PoE* status, configuration, and statistics for the system, slot, and port levels.

Clearing Statistics

You can clear the *PoE* statistics for specified ports or for all ports.

To clear the statistics and reset the counters to 0, enter the following command:

`clear inline-power stats ports [all | port_list]`

Displaying System Power Information

You can display the status of the inline power for the system and, for additional information, display the power budget of the switch.

Display System PoE Status

- Display the *PoE* status for the switch.

  ```
  show inline-power [fast | perpetual]
  ```

  The command provides status for the following areas:

  - Configured inline power status—The status of the inline power for the switch: enabled or disabled.
  - System power surplus—The surplus amount of power on the system, in watts, available for budgeting.
  - Redundant power surplus—The amount of power on the system, in watts, available for budgeting if one power supply is lost.
  - System power usage threshold—The configured power usage threshold for each slot, shown as a percentage of budgeted power. After this threshold has been passed on any slot, the system sends an *SNMP* event and logs a message.
  - Disconnect precedence—The method of denying power to PDs if the budgeted power on any slot is exceeded.
  - Legacy mode—The status of the legacy mode, which allows detection of non-standard PDs.

  The output indicates the following inline power status information for each slot:

  - Inline power status—The status of inline power. The status conditions are:
  - Enabled
  - Disabled
  - Firmware status—The operational status of the slot. The status conditions are:
  - Operational
  - Not operational
  - Disabled
  - Subsystem failure
  - Card not present
  - Slot disabled
  - Budgeted power—The amount of inline power, in watts, that is reserved and available to the slot.
  - Measured power—The amount of power, in watts, that is currently being used by the slot.

  PoE optimization options:

  - Fast—Specifies showing fast PoE power status, which is the ability to provide power to devices at the time of switch power on without waiting for boot up based on last saved PoE state.
  - Perpetual—Specifies showing perpetual PoE power status, which is the ability to preserve PoE power delivery to devices during reboot. Perpetual PoE is a switch-wide setting.

Displaying PoE Status and Statistics on Stand-alone Switches

To display the *PoE* statistics for the switch, run the `show inline-power stats` command.

The command provides the following information:

- Firmware status—Displays the firmware state:
  - Operational
  - Not operational
  - Disabled
  - Subsystem failure
- Firmware revision—Displays the revision number of the PoE firmware.
- Total ports powered—Displays the number of ports powered on specified slot.
- Total ports awaiting power—Displays the number of remaining ports in the slot that are not powered.
- Total ports faulted—Displays the number of ports in a fault state.
- Total ports disabled—Displays the number of ports in a disabled state.

Displaying Port PoE Information

You can display the *PoE* configuration, status, and statistics per port.

Displaying Port PoE Configuration

To display *PoE* configuration for each port, use the following command:

```
show inline-power configuration ports {port_list}
```

This command provides the following information:

- Config—Indicates whether the port is enabled to provide inline power:
  - Enabled: The port can provide inline power.
  - Disabled: The port cannot provide inline power.
- Operator Limit—Displays the configured limit, in milliwatts, for inline power on the port.
- Label—Displays a text string, if any, associated with the port (15 characters maximum).
- Priority – Displays the configured PoE priority of the port.
- Detection – Displays the configured detection of the PoE Port
- Classification – Displays the configured classification of the PoE Port
- Fast – Displays the configured status of fast PoE of the PoE Port

Display Port PoE Status

To display the *PoE* status per port, run the `show inline-power info {detail} ports {port_list}` command. This command provides the following information:

- State—Displays the port power state:
  - Disabled
  - Searching
  - Delivering
  - Faulted
  - Disconnected
  - Other
  - Denied
- PD's power class—Displays the class type of the connected PD:
  - "-----": disabled or searching
  - "class0": class 0 device
  - "class1": class 1 device
  - "class2": class 2 device
  - "class3": class 3 device
  - "class4": class 4 device
  - "class5": class 5 device
  - "class6": class 6 device
  - "class7": class 7 device
  - "class8": class 8 device
- Volts—Displays the measured voltage. A value from 0 to 2 is valid for ports that are in a searching or discovered state.
- Curr—Displays the measured current, in milliamperes, drawn by the PD.
- Power—Displays the measured power, in watts, supplied to the PD.
- Fault—Displays the fault value:
  - None
  - UV/OV fault
  - UV/OV spike
  - Over current
  - Overload
  - Undefined
  - Underload
  - HW fault
  - Discovery resistance fail
  - Operator limit violation
  - Disconnect
  - Discovery resistance, A2D failure
  - Classify, A2D failure

- Sample, A2D failure
- Device fault, A2D failure
- Force on error

The detail command option lists all inline power information for the selected ports. Detail output displays the following information:

- Configured admin state
- Inline power state
- MIB detect status
- Label
- Operator limit
- PD class
- Max allowed power
- Measured power
- Line voltage
- Current
- Fault status
- Detailed status
- Priority

```
* (CIT_32.1.1.6) 5720-24MW-SwitchEngine.1 # show inline-power perpetual
Perpetual PoE
-------------
Disabled
```

To display PoE fast information by port, run the `show inline-power` **`fast ports`** {*`port_list`*} command. This command displays the following information:

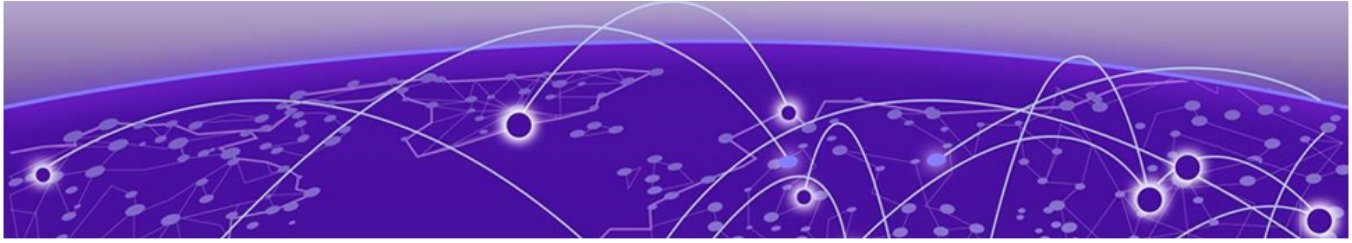
- Port status (enabled/disabled)
- Operator limits
- Priority
- Detection method

```
* (CIT_32.1.0.383) 5720-24MW-SwitchEngine.2 # sh inline-power fast
State   Enable     Disconnect        Last Fast Boot Status
-----   --------   ---------------   ------------------------
HW      Disabled   Deny-port         Succeeded
Cfg     Disabled   Deny-port

State: (HW) Hardware state per-slot, (Cfg) Configured state per-slot.
Fast Boot Status:
  Succeeded - Fast PoE boot succeeded and configuration matches.
  Disabled  - Fast PoE was disabled in hardware at boot time.
  Mismatch  - Fast PoE booted but the configuration in hardware
              did not match the configuration file.
  Failed    - Fast PoE boot failed and PoE was reset.
* (CIT_32.1.0.383) 5720-24MW-SwitchEngine.3
```

Displaying Port PoE Statistics

To display the *PoE* statistics for each port, run the `show inline-power stats ports {port_list}` command. The command provides the following information:

- State—Displays the port power state:
  - Disabled
  - Searching
  - Delivering
  - Faulted
  - Disconnected
  - Other
  - Denied
- PD's power class—Displays the class type of the connected PD:
  - "-----": disabled or searching
  - "class0": class 0 device
  - "class1": class 1 device
  - "class2": class 2 device
  - "class3": class 3 device
  - "class4": class 4 device
  - "class5": class 5 device
  - "class6": class 6 device
  - "class7": class 7 device
  - "class8": class 8 device
- Absent—Displays the number of times the port was disconnected.
- InvSig—Displays the number of times the port had an invalid signature.
- Denied—Displays the number of times the port was denied.
- Over-current—Displays the number of times the port entered an overcurrent state.
- Short—Displays the number of times the port entered undercurrent state.



Policy Manager

Policy Manager and Policies Overview on page 1448
Creating and Editing Policies on page 1448
Applying Policies on page 1451

This chapter provides information about how ExtremeXOS implements policy statements. It includes an overview of the Policy Manager, as well as specific information about how to create, edit, check, and apply policies.

Policy Manager and Policies Overview

One of the processes that make up the ExtremeXOS system is the policy manager. The policy manager is responsible for maintaining a set of policy statements in a policy database and communicating these policy statements to the applications that request them.

Policies are used by the routing protocol applications to control the advertisement, reception, and use of routing information by the switch. Using policies, a set of routes can be selectively permitted (or denied) based on their attributes, for advertisements in the routing domain. The routing protocol application can also modify the attributes of the routing information, based on the policy statements.

Policies are also used by the *ACL (Access Control List)* application to perform packet filtering and forwarding decisions on packets. The ACL application will program these policies into the packet filtering hardware on the switch. Packets can be dropped, forwarded, moved to a different *QoS (Quality of Service)* profile, or counted, based on the policy statements provided by the policy manager.

Creating and Editing Policies

A policy is created by writing a text file that contains a series of rule entries describing match conditions and actions to take.

Policies are created by writing a text file on a separate machine and then downloading it to the switch. Once on the switch, the file is then loaded into a policy database to

be used by applications on the switch. Policy text files can also be created and edited directly on the switch.

Note
Although ExtremeXOS does not prohibit mixing *ACL* and routing type entries in a policy file, it is best to use separate policy files for ACL and routing policies instead of mixing the entries.

When you create a policy file, name the file with the policy name that you will use when applying the policy, and use " .pol" as the filename extension. For example, the policy name "boundary" refers to the text file "boundary.pol".

Using the Edit Command

A VI-like editor is available on the switch to edit policies. There are many commands available with the editor. For information about the editor commands, use any tutorial or documentation about VI. The following is only a short introduction to the editor.

The editor operates in one of two modes: command and input.

1. To edit a policy file on the switch by launching the editor, enter the command:

   `edit policy` *`filename`*.pol

   When a file first opens, you are in the command mode.
2. To write in the file, use the keyboard arrow keys to position your cursor within the file, then press one of the following keys to enter input mode:
   a. **[i]**—Inserts text ahead of the initial cursor position.
   b. **[a]**—Appends text after the initial cursor position.
3. To escape the input mode and return to the command mode, press the **[Esc]** key.

Several commands can be used from the command mode. The following commands are the most commonly used:

- **[dd]**—Deletes the current line.
- **[yy]**—Copies the current line.
- **[p]**—Pastes the line copied.
- **[:w]**—Writes (saves) the file.
- **[:q]**—Quits the file if no changes were made.
- **[:q!]**—Forcefully quits the file without saving changes.
- **[:wq]**—Writes and quits the file.

Using a Separate Machine to Edit Policies

You can also edit policies on a separate machine. Any common text editor can be used to create a policy file. The file is then transferred to the switch using TFTP and then applied.

To transfer policy files to the switch, enter the command:

```
tftp [ ip-address | host-name ] { -v vr_name } { -b block_size } [ -
g | -p ] [ -l local-file { -r remote-file } | -r remote-file { -l
localfile } ]
```

Checking Policies

A policy file can be checked to see if it is syntactically correct. This command can only determine if the syntax of the policy file is correct and can be loaded into the policy manager database. Since a policy can be used by multiple applications, a particular application may have additional constraints on allowable policies.

To check the policy syntax, enter the command:

```
check policy policy_name
```

Refreshing Policies

When a policy file is changed (such as adding, deleting an entry, adding/deleting/modifying a statement), the information in the policy database does not change until the policy is refreshed. The user must refresh the policy so that the latest copy of policy is used. When the policy is refreshed, the new policy file is read, processed, and stored in the server database.

Any clients that use the policy are updated.

- To refresh the policy, enter the command:

  ```
  refresh policy policy_name
  ```

  For *ACL* policies only, during the time that an ACL policy is refreshed, packets on the interface are blackholed, by default. This is to protect the switch during the short time that the policy is being applied to the hardware. It is conceivable that an unwanted packet could be forwarded by the switch as the new ACL is being set up in the hardware. You can disable this behavior.

  Note
  Performing a refresh on multiple ports requires the original and modified policy to coexist at the same time in the intermittent state. If this is not possible due to slice limitations, the refresh will fail with "ACL slice full" error.

- To control the behavior of the switch during an ACL refresh, enter the commands:

  ```
  enable access-list refresh blackhole
  ```

  ```
  disable access-list refresh blackhole
  ```

  In releases previous to ExtremeXOS 11.4, when ACLs were refreshed, all the ACL entries were removed, and new ACL entries were created to implement the newly applied policy.

  Beginning in release 11.4, the policy manager uses Smart Refresh to update the ACLs. When a change is detected, only the ACL changes needed to modify the ACLs are sent to the hardware, and the unchanged entries remain. This behavior avoids having to blackhole packets because the ACLs have been momentarily cleared. Smart Refresh works well up for up to 200 changes. If the number of changes

exceeds 200, you will see this message: Policy file has more than 200 new rules. Smart refresh can not be carried out. Following this message, you will see a prompt based on the current blackhole configuration. If blackhole is disabled you will see the following prompt:

```
Note, the current setting for Access-list Refresh Blackhole is Disabled. WARNING: If
a full refresh is performed, it is possible packets that should be denied may be
forwarded through the switch during the time the access list is being installed.
Would you like to perform a full refresh?
```

If blackhole is enabled, you will see the following prompt:

```
Note, the current setting for Access-list Refresh Blackhole is Enabled.
Would you like to perform a full refresh?
```

Note

ACL refresh may take additional time if the ACL is applied on multiple VLANs with several ACL slices already in the filled state. Additionally, in SummitStack, ACL refresh happens sequentially, such that after successful installation on one node, it will be applied to other nodes one-by-one, causing a slight delay in refresh operation.

To take advantage of Smart Refresh, disable access-list refresh blackholing.

Applying Policies

ACL policies and routing policies are applied using different commands.

Applying ACL Policies

A policy intended to be used as an *ACL* is applied to an interface, and the CLI command option is named *aclname*.

Supply the policy name in place of the *aclname* option.

- To apply an ACL policy, enter the command:
  `configure access-list` *aclname* [**any** | **ports** *port_list*] {**ingress** | **egress**}

  When you use the any keyword, the ACL is applied to all the interfaces and is referred to as the wildcard ACL. This ACL is evaluated for any ports without specific ACLs, and it is also applied to any packets that do not match the specific ACLs applied to the interfaces.

  When an ACL is already configured on an interface, the command is rejected and an error message is displayed.
- To remove an ACL from an interface, enter the command:
  `unconfigure access-list` *policy_name* {**any** | **ports** *port_list* } {**ingress** | **egress**}
- To display the interfaces that have ACLs configured and the ACL that is configured on each. enter the command:
  `show access-list` {**any** | **ports** *port_list* } {**ingress** | **egress**}

Applying Routing Policies

To apply a routing policy, use the command appropriate to the client. Different protocols support different ways to apply policies, but there are some generalities.

Commands that use the keyword import-policy are used to change the attributes of routes installed into the switch routing table by the protocol. These commands cannot be used to determine the routes to be added to the routing table. The following are examples for the *BGP (Border Gateway Protocol)* and *RIP (Routing Information Protocol)* protocols:

```
configure bgp import-policy [policy-name | none]

configure rip import-policy [policy-name | none]
```

Commands that use the keyword route-policy control the routes advertised or received by the protocol. For BGP and RIP, here are some examples:

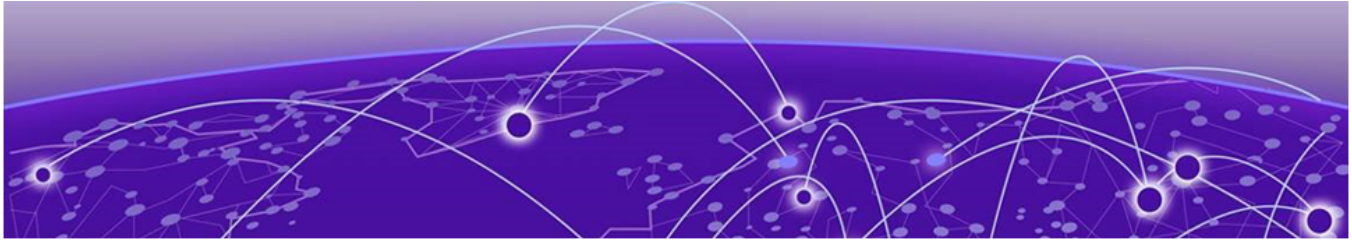
```
configure bgp neighbor [remoteaddr | all] {address-family [ipv4-unicast
| ipv4-multicast |ipv6-unicast | ipv6-multicast |vpnv4]} route-policy
[in | out] [none | policy]

configure bgp peer-group peer-group-name {address-family [ipv4-unicast |
ipv4-multicast |ipv6-unicast | ipv6-multicast |vpnv4]} route-policy [in
|out] [none | policy]

configure rip vlan [vlan_name | all] route-policy [in | out] [policy-
name | none]
```

Other examples of commands that use routing policies include:

```
configure ospf area area-identifier external-filter [policy-map |none]

configure ospf add vlan [vlan-name | all] area area-identifier {passive}
{vr vrf_name}

configure rip vlan [vlan_name | all] trusted-gateway [policy-name |
none]
```

To remove a routing policy, use the **none** option in the command.

Note
When exporting a route into BGP with an export policy, the AS-Path is limited to a maximum of 15 AS numbers.



Quality of Service

QoS Overview on page 1454
Applications and Types of QoS on page 1455
Traffic Groups on page 1457
Introduction to Rate Limiting, Rate Shaping, and Scheduling on page 1462
Introduction to WRED on page 1465
Meters on page 1469
QoS Profiles on page 1469
Class of Service (CoS) on page 1472
Configuring QoS on page 1474
Displaying QoS Configuration and Performance on page 1487

VXLAN gateway uses a DSCP value of 0 when the tenant packet is not IP. Otherwise, it copies the tenant DSCP value to the outer IP header. In order to overwrite this, user must configure an *ACL (Access Control List)* rule. This ACL is general and applied to all virtual networks. The following is an example of this ACL rule:

```
entry map {
      if {
          protocol udp;
          destination-port 4789;
      }
      then {
          qosprofile qp3;
          replace-dscp;
          replace-dot1p;
 }
 }
```

Then, the following command is used to put the outer DSCP value to a fixed value of 32:

```
configure diffserv replacement qp3 code-point 32
```

As indicated in the example above, the dot1p value can also be replaced. In the absence of the ACL, VXLAN gateway copies the tenant dot1p value to the network. The replacement happens on the outer headers as the example above illustrates. *QoS (Quality of Service)* examination (dot1p or diffserv) happens on the first header. On the tenant port, it is examining the ethernet and IP header of tenant frames. On the network port, it is examining the outer Ethernet and IP header. There is no change to the existing support for treatment of packets, such as rate-limiting and queuing.

QoS Overview

Quality of Service (QoS) is a feature that allows you to configure a switch to provide different levels of service to different groups of traffic. For example, QoS allows you to do the following:

- Give some traffic groups higher priority access to network resources.
- Reserve bandwidth for special traffic groups.
- Restrict some traffic groups to bandwidth or data rates defined in a Service Level Agreement (SLA).
- Count frames and packets that exceed specified limits and optionally discard them (rate limiting).
- Queue or buffer frames and packets that exceed specified limits and forward them later (rate shaping).
- Modify QoS related fields in forwarded frames and packets (remarking).

Figure 180 shows the QoS components that provide these features on Extreme Networks switches.

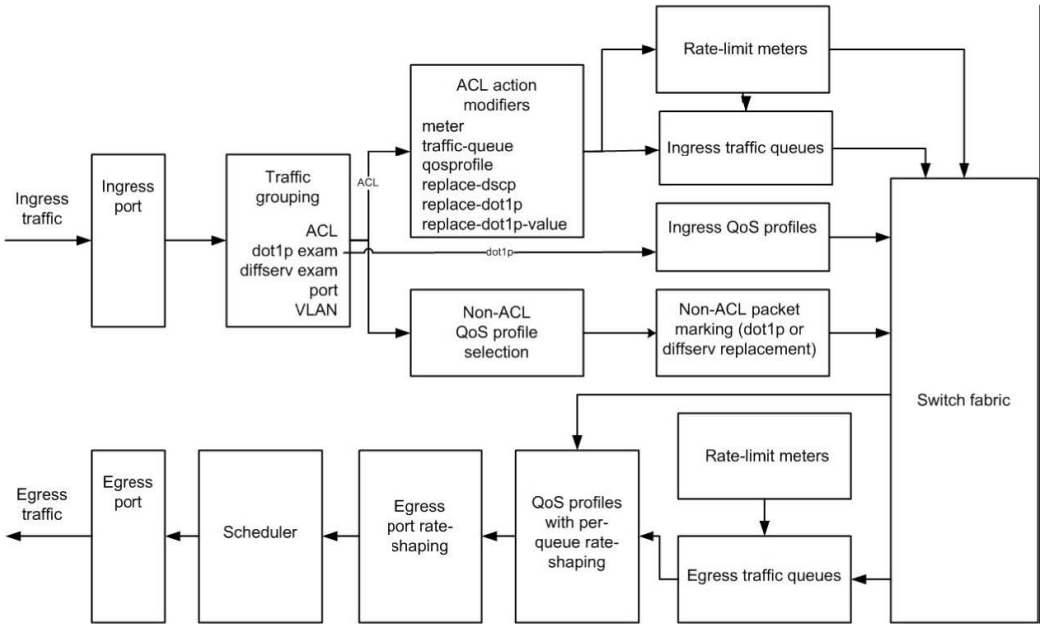


Figure 180: QoS on Extreme Networks Switches

Data enters the ingress port and is sorted into traffic groups, which can be classified as either ACL-based or non-ACL-based.

The ACL-based traffic groups provide the most control of QoS features and can be used to apply ingress and egress rate limiting and rate shaping as follows:

- Subject ingress traffic to rate limit meters
- Specify ingress hardware queues (QoS profiles) for rate limiting and rate shaping
- Specify ingress software traffic queues for rate limiting and rate shaping (these can be associated with egress traffic queues for additional QoS control)

- Specify egress software traffic queues for rate limiting and rate shaping
- Specify egress QoS profiles for rate limiting and rate shaping
- Change the dot1p or Differential Services (DiffServ) values in egress frames or packets

Non-ACL-based traffic groups specify an ingress or egress QoS profile for rate limiting and rate shaping. These groups cannot use ingress or egress software traffic queues. However, non-ACL-based traffic groups can use the packet marking feature to change the dot1p or DiffServ values in egress frames or packets.

The ingress rate-limiting and rate-shaping features allow you to apply QoS to incoming traffic before it reaches the switch fabric. If some out-of-profile traffic needs to be dropped, it is better to drop it before it consumes resources in the switch fabric.

All ingress traffic is linked to an egress traffic queue or QoS profile before it reaches the switch fabric. This information is forwarded with the traffic to the egress interface, where it selects the appropriate egress traffic queue or QoS profile. Egress traffic from all traffic queues and QoS profiles is forwarded to the egress port rate-shaping feature, which applies QoS to the entire port. When multiple QoS profiles are contending for egress bandwidth, the scheduler determines which queues are serviced.

The following sections provide more information on QoS.

Applications and Types of QoS

Different applications have different *QoS* requirements. Table 126 on page 1455 summarizes the QoS guidelines for different types of network traffic.

Table 126: Traffic Type and QoS Guidelines

Traffic Type	Key QoS Parameters
Voice	Minimum bandwidth, priority
Video	Medium bandwidth, priority, buffering (varies)
Database	Minimum bandwidth
Web browsing	Minimum bandwidth for critical applications, maximum bandwidth for noncritical applications
File server	Minimum bandwidth

Consider these parameters as general guidelines and not as strict recommendations. After QoS parameters have been set, you can monitor the performance of the application to determine if the actual behavior of the applications matches your expectations. It is very important to understand the needs and behavior of the particular applications you want to protect or limit. Behavioral aspects to consider

include bandwidth needs, sensitivity to latency and jitter, and sensitivity and impact of packet loss.

Note
Full-duplex links should be used when deploying policy-based QoS. Half-duplex operation on links can make delivery of guaranteed minimum bandwidth impossible.

Voice Applications

Voice applications, or voice over IP (VoIP), typically demand small amounts of bandwidth. However, the bandwidth must be constant and predictable because voice applications are typically sensitive to latency (inter-packet delay) and jitter (variation in inter-packet delay). The most important *QoS* parameter to establish for voice applications is minimum bandwidth, followed by priority.

Video Applications

Video applications are similar in needs to voice applications, with the exception that bandwidth requirements are somewhat larger, depending on the encoding. It is important to understand the behavior of the video application being used. For example, in the playback of stored video streams, some applications can transmit large amounts of data for multiple streams in one spike, with the expectation that the end stations will buffer significant amounts of video-stream data. This can present a problem to the network infrastructure, because the network must be capable of buffering the transmitted spikes where there are speed differences (for example, going from gigabit Ethernet to Fast Ethernet). Key *QoS* parameters for video applications include minimum bandwidth and priority, and possibly buffering (depending upon the behavior of the application).

Critical Database Applications

Database applications, such as those associated with Enterprise Resource Planning (ERP), typically do not demand significant bandwidth and are tolerant of delay. You can establish a minimum bandwidth using a priority less than that of delay-sensitive applications.

Web Browsing Applications

QoS needs for web browsing applications cannot be generalized into a single category. For example, ERP applications that use a browser front-end might be more important than retrieving daily news information. Traffic groupings can typically be distinguished from each other by their server source and destinations. Most browser-based applications are distinguished by the dataflow being asymmetric (small dataflows from the browser client, large dataflows from the server to the browser client).

An exception to this might be created by some Java™-based applications. In addition, web-based applications are generally tolerant of latency, jitter, and some packet loss;

however, small packet loss might have a large impact on perceived performance because of the nature of TCP. The relevant parameter for protecting browser applications is minimum bandwidth. The relevant parameter for preventing non-critical browser applications from overwhelming the network is maximum bandwidth.

File Server Applications

With some dependencies on the network operating system, file serving typically poses the greatest demand on bandwidth, although file server applications are very tolerant of latency, jitter, and some packet loss, depending on the network operating system and the use of TCP or UDP.

Traffic Groups

A traffic group defines the ingress traffic to which you want to apply some level of *QoS*. You can use the ExtremeXOS software to define traffic groups based on the following:

- Frame or packet header information such as IP address or MAC address
- *CoS (Class of Service)* 802.1p bits in the frame header
- DiffServ information in a packet header
- Ingress port number
- *VLAN (Virtual LAN)* ID

Traffic groups that are defined based on frame or packet information are usually defined in Access Control Lists (ACLs). The exception to this rule is the CoS and DiffServ information, which you can use to define traffic groups without ACLs.

The function of the CoS and DiffServ traffic groups is sometimes referred to as explicit packet marking, and it uses information contained within a frame or packet to explicitly determine a class of service. An advantage of explicit packet marking is that the class of service information can be carried throughout the network infrastructure, without repeating what can be complex traffic group policies at each switch location. Another advantage is that end stations can perform their own packet marking on an application-specific basis. Extreme Networks switch products have the capability of observing and manipulating packet marking information with no performance penalty.

The CoS and DiffServ capabilities (on supported platforms) are not impacted by the switching or routing configuration of the switch. For example, 802.1p information can be preserved across a routed switch boundary and DiffServ code points can be observed or overwritten across a Layer 2 switch boundary.

During QoS configuration, you configure the QoS level first by configuring QoS profiles, traffic queues, and meters, and then you define a traffic group and assign the traffic group to the QoS configuration.

ACL-Based Traffic Groups

An *ACL*-based traffic group allows you to use ACL rules in an ACL policy file to define the traffic to which you want to apply *QoS*. An ACL-based traffic group requires more effort to create, but the ACL rules give you more control over which traffic is selected for

the traffic group. For example, you can use an ACL to add traffic to a traffic group based on the following frame or packet components:

- MAC source or destination address
- Ethertype
- IP source or destination address
- IP protocol
- TCP flag
- TCP, UDP, or other Layer 4 protocol
- TCP or UDP port information
- IP fragmentation

Depending on the platform you are using, traffic in an ACL traffic group can be processed as follows:

- Assigned to an ingress meter for rate limiting
- Marked for an egress QoS profile for rate shaping
- Marked for an egress traffic queue for rate shaping
- Marked for DSCP replacement on egress
- Marked for 802.1p priority replacement on egress
- Assigned to an egress meter for rate limiting

When you are deciding whether to use an ACL-based traffic group or another type of traffic group, consider what QoS features you want to apply to the traffic group. Some QoS features can only apply to ACL-based traffic groups.

Note
ACLs are discussed in detail in the ACLs chapter.

CoS 802.1p-Based Traffic Groups

CoS 802.1p-based traffic groups forward traffic to *QoS* features based on the three 802.1p priority bits in an Ethernet frame. The 802.1p priority bits are located between the 802.1Q type field and the 802.1Q *VLAN* ID as shown in Figure 181.

Typically CoS will be used in conjunction with policy, however it may also work independently. In a policy-enabled system, policy may use any CoS index setting (0-255) to configure ACLs to assign a qosprofile, meter, dot1p replacement and TOS-rewrite value. By default, 802.1p priorities 0-7 are mapped to CoS indices 0-7.

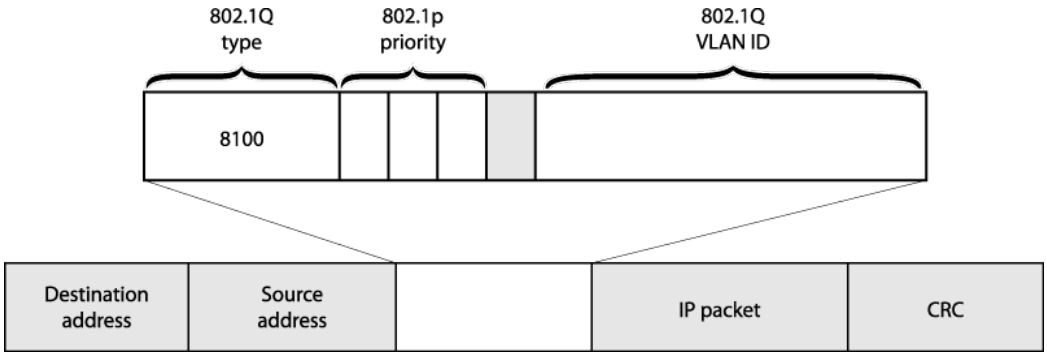


Figure 181: 802.1p Priority Bits

The three 802.1p priority bits define up to eight traffic groups that are predefined in the ExtremeXOS software.

On SummitStack and ExtremeSwitching series switches, the traffic groups direct traffic to egress QoS profiles for egress rate shaping (see the following table).

You do not need to define 802.1p-based traffic groups. You can enable or disable the use of these traffic groups by enabling or disabling the 802.1p examination feature. You can also configure which 802.1p values map to which QoS profiles.

A related feature is the 802.1p replacement feature, which allows you to configure the software to replace the 802.1p bits in an ingress frame with a different value in the egress frame. For more information on 802.1p replacement, see Configuring 802.1p or DSCP Replacement on page 1477.

DiffServ-Based Traffic Groups

DiffServ-based traffic groups forward traffic to egress *QoS* profiles based on the type-of-service (TOS) or traffic class (TC) information in an IP packet. In many systems, this TOS or TC information is replaced with a DiffServ field that uses six of the eight bits for a *DSCP (Differentiated Services Code Point)*, as shown in Figure 182. (The other two bits are not used.)

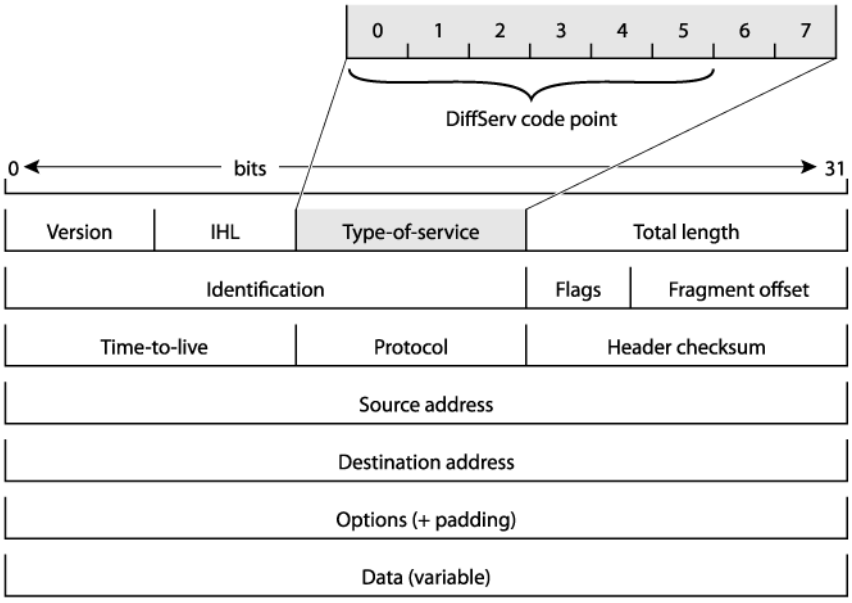


Figure 182: DiffServe Code Point

Because the DSCP uses six bits, it has 64 possible values (2^6 = 64). By default, the values are grouped and assigned to the default QoS profiles as listed in Table 127.

Table 127: Default DSCP-to-QoS-Profile Mapping

Traffic Group Code Point	SummitStack and ExtremeSwitching Switches QoS Profile
0-7	QP1
8-15	QP1
16-23	QP1
24-31	QP1
32-39	QP1
40-47	QP1
48-55	QP1
56-63	QP8

Note

The default DiffServ examination mappings apply on ports in more than one virtual router. If you attempt to configure DiffServe examination or replacement on a port that is in more than one virtual router, the system returns the following message:

```
Warning: Port belongs to more than one VR. Port properties related to diff serv
and code replacement will not take effect.
```

You do not need to define these traffic groups. You can enable or disable the use of these traffic groups by enabling or disabling the DiffServ examination feature as

described in Configuring a DiffServ-Based Traffic Group on page 1483. You can also configure which DSCP values map to which queues.

Note
When DiffServ examination is enabled on 1 Gigabit Ethernet ports, 802.1p replacement is enabled and cannot be disabled. The ingress 802.1p value is replaced with the 802.1p value assigned to the egress QoS profile.

A related feature is the DiffServ replacement feature, which allows you to configure the software to replace the DSCP in an ingress frame with a different value in the egress frame. For more information on DiffServ replacement, see Configuring 802.1p or DSCP Replacement on page 1477.

Port-Based Traffic Groups

Port-based traffic groups forward traffic to egress *QoS* profiles based on the incoming port number. There are no default port-based traffic groups; you must configure each port-based traffic group.

Note
Port-based traffic groups apply to all packets.

VLAN-Based Traffic Groups

VLAN-based traffic groups forward traffic to egress *QoS* profiles based on the VLAN membership of the ingress port. There are no default VLAN-based traffic groups; you must configure each VLAN-based traffic group.

Note
VLAN-based traffic groups apply to all packets.

Precedence of Traffic Groups

The ExtremeXOS software allows you to define multiple traffic groups, and you can configure traffic groups in such a way that multiple traffic groups apply to an ingress frame or packet. When an ingress frame or packet matches two or more traffic groups, the software chooses one traffic group based on the precedence defined for the switch platform. In general, the more specific traffic group definition takes precedence. Table

128 on page 1462 shows the traffic group precedence for the supported switch platforms (number 1 is the highest precedence).

Note
Beginning in ExtremeXOS 16.1, the port, *VLAN* and DiffServ-based traffic groups will only work if 802.1p examination is disabled.

Table 128: Traffic Group Precedence

SummitStack and ExtremeSwitching Switches
1. *ACL*-based traffic groups for IP packets (specifies IP address information) 2. ACL-based traffic groups for Ethernet frames (specifies MAC address information) 3. *CoS* 802.1p-based traffic groups 4. Port-based traffic groups 5. VLAN-based traffic groups 6. DiffServ-based traffic groups

Introduction to Rate Limiting, Rate Shaping, and Scheduling

The terms *rate limiting* and *rate shaping* are used throughout this chapter to describe *QoS* features. Some QoS features perform both rate limiting and rate shaping. Rate limiting is the process of restricting traffic to a peak rate (PR). Rate shaping is the process of reshaping traffic throughput to give preference to higher priority traffic or to buffer traffic until forwarding resources become available.

Both rate limiting and rate shaping allow you to take action on traffic that exceeds the configured limits. These actions include forwarding traffic, dropping traffic, and marking the excess traffic for possible drops later in the communication path. Software counters allow you to record traffic statistics such as total packets forwarded and total packets dropped.

Single-Rate QoS

Single-rate *QoS* defines a single rate for traffic that is subject to QoS. Single-rate QoS is the most basic form of rate limiting and is well suited for constant rate traffic flows such as video or where more complex dual-rate QoS is not needed. The traffic that meets the rate requirement is considered in-profile. Traffic that does not meet the specified rate is considered out-of-profile. A two-color system is often used to describe or mark the single-rate QoS result. In-profile traffic is marked green, and out-of-profile traffic is marked red.

Single-rate rate-limiters pass traffic that is in-profile or marked green. Out-of-profile traffic (marked red) is subject to whatever action is configured for out-of-profile traffic. Out of profile traffic can be forwarded if bandwidth is available, dropped, or marked for a possible drop later in the communication path.

All traffic that arrives at or below the PR is considered in-profile and marked green. All traffic that arrives above the PR is considered out-of-profile and marked red. When the

traffic exceeds the capacity of the rate-limiting component, all traffic that is marked red is dropped.

Another type of single-rate QoS is used on SummitStack and ExtremeSwitching series switches. A committed information rate (CIR) establishes a reserved traffic rate, and a peak burst size (PBS) establishes a maximum size for a traffic stream. If a traffic stream is at or below the CIR and the PBS, it is considered to be within profile and marked green. If a traffic stream exceeds either the CIR or the PBS, it is considered out-of-profile and marked red. On these switches, you can configure the single-rate rate-limiting components to drop traffic marked red or set a drop precedence for that traffic. You can also specify a DSCP value to mark the out-of-profile traffic.

Dual-rate QoS

Dual-rate *QoS* defines two rates for traffic that is subject to QoS. The lower of the two rates is the CIR, which establishes a reserved traffic rate. The higher of the two rates is the peak rate, which establishes an upper limit for traffic. Dual-rate QoS is well suited to bursty traffic patterns which are common with typical data traffic. Dual-rate QoS is widely used in legacy Frame Relay and ATM leased lines.

Note
You must configure the peak rate higher than the committed rate.

A three-color system is used with dual-rate QoS. As with single-rate QoS, traffic at or below the CIR is considered in-profile, marked green, and forwarded. The traffic that is above the CIR and below the PIR is out-of-profile and marked yellow. Traffic above the PIR is also out-of-profile and marked red. When incoming traffic is already marked yellow and is out of profile, it is marked red. Different switch platforms take different actions on the traffic marked yellow or red.

Rate Specification Options

The ExtremeXOS software allows you to specify the CIR and PR in gigabits per second (Gbps), megabits per second (Mbps), or kilobits per second (Kbps). Most commands also allow you to specify the CIR and PR as a percentage of the maximum port bandwidth using the **minbw** (CIR) and **maxbw** (PR) options. The default value on all minimum bandwidth parameters is 0%, and the default value on all maximum bandwidth parameters is 100%.

QoS can be applied at different locations in the traffic path using the following rate-limiting and rate-shaping components:

- Ingress meters
- Egress traffic queues
- Egress meters
- Egress *QoS* profiles
- Egress ports

The CIR or minimum bandwidth configuration for a rate-limiting or rate-shaping component is a bandwidth guarantee for that component at a particular location in the traffic path. The guarantees for all components at a specified location should add up to less than 100% and should account for the traffic needs of the other components. For example, if you configure 25% minimum bandwidth for four out of eight queues at a particular location, there will be no available bandwidth for the remaining four queues when traffic exceeds the port capacity. Bandwidth unused by a queue can be used by other queues.

The rate-shaping configuration is configured at the location to which it applies on most platforms.

Disabling Rate Limiting and Rate Shaping

All switch platforms provide multiple *QoS* components in the traffic path that provide rate limiting or rate shaping. These components give you control over where and how the rate shaping is applied. However, your application might not require rate shaping at every component. The default configuration for most components provides no rate shaping. When rate shaping is disabled on a component, the CIR is set to 0 (minbw=0%) and the PR is set to the maximum bandwidth (maxbw=100%). This setting reserves no bandwidth for the component and allows the component to use 100% of the port bandwidth. If you need to remove rate shaping from a QoS component, configure these settings on that component.

Scheduling

Scheduling is the process that determines which traffic is forwarded when multiple *QoS* components are competing for egress bandwidth. The ExtremeXOS software supports the following scheduling methods:

- Strict priority: All higher priority queues are serviced before lower priority queues. This ensures that high priority traffic receives access to available network bandwidth immediately, but can result in lower priority traffic being starved. As long as a queued packet remains in a higher-priority queue, any lower-priority queues are not serviced.
- Weighted round robin: All queues are given access to a relative amount of bandwidth based on the weight assigned to the queue. When you configure a QoS profile with a weight of 4, that queue is serviced four times as frequently as a queue with a weight of 1. The hardware services higher-weighted queues more frequently, but lower-weighted queues continue to be serviced at all times. Initially, the weight for all queues is set to 1, which gives them equal weight. If all queues are set to 4, for example, all queues still have equal weight, but each queue is serviced for a longer period.
- Weighted deficit round robin: All queues are given access based on the configured priority level and a round-robin algorithm.

Scheduling takes place on the egress interface and includes consideration for the color-marking of egress frames and packets. Green-marked traffic has the highest priority and is forwarded based on the scheduling method. When multiple queues are

competing for bandwidth, yellow-marked traffic might be dropped or remarked red. Red-marked traffic is dropped when no bandwidth is available. If yellow-marked traffic is forwarded to the egress port rate-shaping component, it can be dropped there if the egress port is congested.

Introduction to WRED

The weighted random early detection (WRED) feature is supported on some platforms to avoid congestion in traffic queues or *QoS* profiles. WRED improves upon the TCP congestion control mechanism.

The TCP congestion control mechanism on hosts detects congestion when packets are lost and lowers the packet transmission rate in response. At the switch, packets are dropped when a queue is full. When multiple hosts forward packets that are dropped, multiple hosts reduce the transmission rate. This creates a global synchronization problem as multiple hosts overwhelm a queue and then simultaneously lower their transmission rate and under utilize the queue.

WRED is an extension to random early detection (RED), which calculates an average queue size and randomly discards packets in proportion to the queue usage. At low usage levels, no packets are discarded. As the average queue size exceeds configured thresholds, packets are discarded in proportion to the queue usage. Discarding packets early causes some (but not all) hosts to reduce their transmission rate, which reduces queue congestion. The random nature of the discard process reduces the global synchronization problem.

WRED extends RED by applying different discard rules for different types of traffic. Typically, WRED is used on core routers and takes action based on the packet contents established at edge routers. Edge routers can use the IP precedence or DSCP value to mark packets as committed (green), conforming (yellow), or exceeded (red). The marking process and these colors are described in Introduction to Rate Limiting, Rate Shaping, and Scheduling on page 1462.

The ExtremeXOS WRED implementation varies per platform and allows you to configure the following:

- Minimum threshold for dropped packets.
- Maximum threshold for dropped packets.
- Maximum drop rate.
- An average weight control that determines how WRED calculates the average queue size.

WRED does not drop packets at calculated average queue sizes below the minimum threshold (although packets would be dropped if the queue fills).

When the calculated average queue size rises above the maximum threshold, WRED drops packets at the maximum drop rate. When the calculated average falls between the minimum and maximum thresholds, packets are randomly dropped at a proportional rate between zero and the maximum drop rate. As the queue fills, more packets are dropped.

The average weight parameter provides some control over how the average queue size is calculated and the probability of packet drop. Increasing the avg_weight value reduces the probability that traffic is dropped. Conversely, decreasing the avg_weight value increases the probability that traffic is dropped.

On ExtremeSwitching Universal switches, you can configure up to four WRED profiles or configurations per QoS profile, enabling you to create custom WRED configurations for up to 32 traffic flows (4 WRED profiles x 8 QoS profiles).

Each QoS profile supports WRED profiles for the following colors of traffic:

- TCP green
- Non-TCP green
- TCP red
- Non-TCP red

Without support for non-TCP traffic management, non-TCP traffic could monopolize a QoS profile in which TCP traffic is regulated, effectively giving non-TCP traffic priority over TCP traffic. With support for both TCP and non-TCP traffic, WRED allows you to regulate different types of traffic independently, giving you greater control over which type of traffic is dropped first and most frequently.

The typical WRED configuration establishes the lowest probability for packet drop for green traffic, which conforms to established limits along the transmission path. A typical WRED configuration establishes a higher probability for packet drop for red colored traffic, because it has already exceeded established limits earlier in the transmission path. All traffic (green and red) is dropped when the queue is full, so the goal is to configure the WRED settings for each color in such a way as to prevent the queue from filling frequently.

Explicit Congestion Notification (ECN)

Weighted Random Early Detection (WRED) drops packets based on the average length exceeding a specific threshold value to indicate congestion. Explicit Congestion Notification (ECN) is an extension to WRED that marks the drop-eligible packets, instead of dropping, using the same criteria of minimum threshold, maximum threshold, and drop probability. Figure 183 shows the WRED behavior without ECN and with ECN.

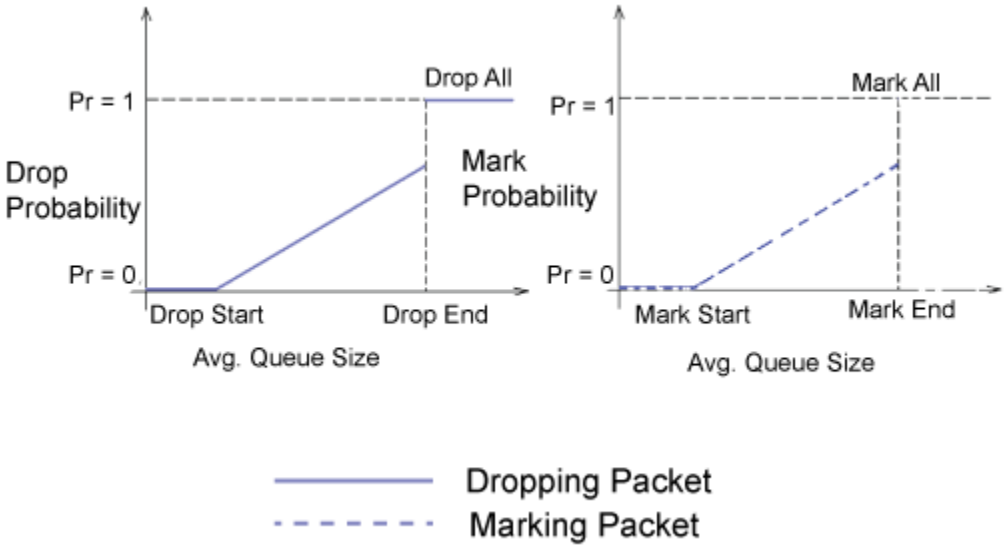


Figure 183: WRED Behavior with and without ECN

The current WRED implementation prevents tail dropping of packets that are queued in the output queue by selectively dropping packets early. This avoids dropping of all packets at once when congestion occurs. With ECN functionality, routers are no longer limited to packet drops to indicate congestion. Routers can instead set the CE (Congestion Experienced) codepoint in the IP header of packets from ECN-capable transport entities. The transport endpoints use the ECE and CWR bits of the TCP header to mitigate congestion.

RFC 3168 states that the two-bit ECN field comprises an ECT bit and a CE bit that are used to implement ECN. The ECT bit and the CE bit can be combined into four possible ECN field values 00 to 11. The first number is ECT bit and the second number is the CE bit. Table 129 shows the possible combination.

Table 129: ENC Field Values

ECT Bit Value	CE Bit Value	Combined Inference
0	0	Not ECN capable. Packet is not using ECN.
0	1	Endpoints are ECN capable. Data senders use these values to indicate that the endpoints are ECN capable. Routers treat the two values identically.
1	0	
1	1	Congestion experienced.

Table 130 indicates the processing applied on incoming packets that are identified as drop-eligible in various ECN scenarios.

Table 130: ECN Processing

Incoming ECN Bits	ECN Configuration	Action if Packet Is Drop-Eligible	Outgoing ECN bits
00 (Non-ECT)	On/Off	Drop.	N/A (packet dropped).
ECT (01 or 10)	Off	Drop.	N/A (packet dropped).
ECT (01 or 10)	On	Not dropped. CE bit is marked.	11 indicates "Congestion Experienced".
CE (11)	Off	Drop.	N/A (packet dropped).
CE (11)	On	Not dropped. Packet already marked, just forward.	11 indicates "Congestion Experienced".

To configure ECN capability as on or off, use the following command:

```
configure qosprofile egress qp_num wred ecn [on | off] ports [port_list | all]
```

To view ECN information, use the following commands:

```
show wredprofile {ports [port_list | all]}
```

```
show ports port_list wred ecn {no-refresh | refresh}
```

To clear the ECN counters, use the following commands:

```
clear counters wred ecn
```

Supported Platforms

This feature is available on all ExtremeSwitching Universal switches.

Limitations

- TCP yellow traffic color is not supported.
- ECN over VXLAN is not supported. Any tunneling protocol implemented with double headers is not supported.
- The ECN counters showing the number of marked packets is not available yet, so those appear as 0.
- This implementation is limited to ECN pass-through. End points having TCP/DCTCP/RoCEv2 capabilities can use the ECN services.

Meters

Meters are used to define ingress rate-limiting and rate-shaping. Some platforms also support meters for egress traffic. The following sections provide information on meters for specific platforms.

The SummitStack and standalone switches use a single-rate meter to determine if ingress traffic is in-profile or out-of-profile.

On all ExtremeSwitching switches, you can also use single-rate meters to determine if egress traffic is in-profile or out-of-profile.

When *ACL* meters are applied to a *VLAN* or to any, the rate limit is applied to each port group. To determine which ports are contained within a port group, use any of the following commands:

```
show access-list usage acl-range port port

show access-list usage acl-rule port port

show access-list usage acl-slice port port
```

The out-of-profile actions are drop, set the drop precedence, mark the dot1p, or mark the DSCP with configured value. Additionally, each meter has an associated out-of-profile counter that counts the number of packets that were above the committed-rate (and subject to the out-of-profile-action).

Prior to ExtremeXOS 21.1, when traffic exceeded the specified meter rate, the traffic could be marked with a user-specified diffserv code point. As of ExtremeXOS 21.1, this out-of-profile marking capability is enhanced to additionally include the option of marking the 802.1p value. The 802.1p value that is marked will be the outer 802.1p value in both VLAN and VMAN tagging scenarios. In the latter, the SVLAN's 802.1p value will be modified. Like the existing dscp out-of-profile meter action, this out-of-profile action will work on global and per-port meters.

On ExtremeSwitching series switches, the meters are a per-chip, per-slice resource (see ACLs for complete information.)

Note
If set-drop-precedence is used as a meter out-of-profile action, you should configure WRED on the egress port so that red packets marked by the meter are dropped in case of egress congestion.

QoS Profiles

QoS profiles are queues that provide ingress or egress rate limiting and rate shaping. The following sections provide more information on QoS profiles.

Egress Qos Profiles

Egress *QoS* profiles are supported on all ExtremeXOS switches and allow you to provide dual-rate egress rate-shaping for all traffic groups on all egress ports. Any configuration you apply to an egress QoS profile is applied to the same egress QoS profile on all other ports, unless a QoS profile parameter has been overridden for a port.

When you are configuring *ACL*-based traffic groups, you can use the qosprofile action modifier to select an egress QoS profile. For DiffServ-, port-, and *VLAN*-based traffic groups, the traffic group configuration selects the egress QoS profile. For *CoS* dot1p traffic groups on all platforms, the dot1p value selects the egress QoS profile.

Egress QoS profile operation depends on the switch type and is described in the following sections.

SummitStack and ExtremeSwitching series switches have two default egress QoS profiles named QP1 and QP8. You can configure up to six additional QoS profiles (QP2 through QP7) on the switch. However, on a SummitStack, you cannot create QoS profile QP7, as this profile is reserved for system control traffic. The default settings for egress QoS profiles are summarized Table 131 on page 1470.

Table 131: Default QoS Profile Parameters on all Platforms

Ingress 802.1p Priority Value	Egress QoS Profile Name	Queue Service Priority Value	Buffer	Weight	Notes
0-6	QP1	1 (Low)	100%	1	This QoS profile is part of the default configuration and cannot be deleted.
	QP2	2 (LowHi)	100%	1	You must create this QoS profile before using it.
	QP3	3 (Normal)	100%	1	You must create this QoS profile before using it.
	QP4	4 (NormalHi)	100%	1	You must create this QoS profile before using it.
	QP5	5 (Medium)	100%	1	You must create this QoS profile before using it.
	QP6	6 (MediumHi)	100%	1	You must create this QoS profile before using it.
	QP7	7 (High)	100%	1	You must create this QoS profile before using it. You cannot create this QoS profile on SummitStack.
7	QP8	8 (HighHi)	100%	1	This QoS profile is part of the default configuration and cannot be deleted.

[9] The QoS Profile Name cannot be changed.
[10] The Queue Service Priority value cannot be changed.

For CoS 802.1p traffic groups, the ingress 802.1p priority value selects a specific QoS profile as shown in Table 131. This mapping can be changed as described in Changing the 802.1p Priority to QoS Profile Mapping on page 1483. For traffic groups other than 802.1p-based groups, the traffic group configuration selects a specific egress QoS profile by name.

The default dual-rate QoS configuration is 0% for minimum bandwidth and 100% for maximum bandwidth.

The QoS profile for each port receives a default buffer reservation. All unreserved buffer space is part of a buffer pool, which can be used by QoS profiles when reserved space runs out, provided that the configuration for that QoS profile and port allows it.

You can increase the size of the shared buffer pool by reducing the global buffer reservation for a QoS profile on all switch ports. You can restrict buffer usage for a QoS profile in amounts ranging from 1 to 100%, in whole integers.

You can also override the global buffer reservation to increase or decrease the buffer space allotment for a specific QoS profile on one or more ports. Using the buffer override feature, you can override the global setting to use from 1–10,000% of the configured global allotment. The system does not drop any packets as long as reserved packet buffer memory for the port and QoS profile or shared packet memory for the port (`configure port` *`port_list`* **`shared-packet-buffer`** command) remains available.

Note

In a SummitStack, the scheduling algorithm is automatically programmed by ExtremeXOS for the stacking links only, and might be different from the algorithm you select.

Use of all eight queues on all ports can result in insufficient buffering to sustain zero packet loss throughput during full-mesh connectivity with large packets.

When multiple QoS profiles are contending for port bandwidth and the egress traffic in each profile is within profile, the scheduler determines how the QoS profiles are serviced as described in Scheduling on page 1464. In strict-priority mode, the queues are serviced based on the queue service priority value. In weighted fair-queuing mode, the queues are serviced based on the configured weight.

When configured to do so, the priority of a QoS profile can determine the 802.1p bits used in the priority field of a forwarded frame (see Introduction to Rate Limiting, Rate Shaping, and Scheduling on page 1462). The priority of a QoS profile can determine the DiffServ code point value used in an IP packet when the packet is forwarded (see Replacement of DSCP on Egress on page 1478).

A QoS profile change does not alter the behavior of the switch until it is assigned to a traffic group.

Egress Port Rate Limiting and Rate Shaping

Egress port rate limiting and rate shaping allow you to define limits for all egress traffic coming from the egress *QoS* profiles and traffic queues.

You can apply single-rate rate-limiting and control shared packet buffer space (`configure port port_list` **`shared-packet-buffer`** command) for each port.

You can also configure ports to pass an unlimited flow as described in Disabling Rate Limiting and Rate Shaping on page 1464.

Configuring Overhead-Bytes in Calculating Rate-Limiting and Rate-Shaping

You can configure the size of overhead-bytes used in the calculation of rate-limiting and rate-shaping. Overhead-bytes include size of preamble and inter-frame gap. In releases earlier than 21.1, ExtremeXOS included these overhead-bytes during traffic rate-limit and shaping techniques. Now, you can configure overhead-bytes using the `configure forwarding rate-limit overhead-bytes <bytes>` command.

For example, if the frame size is 68 bytes configuring overhead-bytes of 0, the switch calculates rate-limit and shaping techniques with 68 bytes as the frame size. If 20 is the configured overhead-bytes size, the switch calculates rate-limit and shaping techniques with 88 bytes as the frame size. This configuration is applicable for ingress meters, egress *QoS* profile, egress port rate-limiting, and rate-shaping configurations.

Class of Service (CoS)

CoS provides a means to configure 802.1p-based traffic grouping, *QoS* rate-shaping and scheduling, ingress and egress metering and flood rate-limiting via *SNMP (Simple Network Management Protocol)*. The CoS MIB is fully integrated with ONEPolicy to provide a solution to management entities.

Class of Service (CoS) Settings

Each of the 256 *CoS* entries, indexed 0-255, are mapped to a 802.1p priority (0-7), ToS value (0-255), virtual txq-reference (0-15) and virtual irl-reference (0-31). A virtual reference is used to abstract the hardware capabilities from the CoS row. This allows for the management entity to create user CoS profiles based on the user and not multiple user profiles for each hardware platform. The virtual references are then mapped to actual hardware resources. On ExtremeXOS, the txq-reference mappings 0-7 are fixed to *QoS* profiles 1-8 and txq-reference 8-15 map to QoS profile 8. The irl-reference mappings are also fixed to the per-port ingress meters. This mapping is dependent on the number of per-port ingress meters supported on the platform.CoS entries 0-7 are equivalent to the dot1p traffic groupings for priorities 0-7.

Port Groups

Port groups allow you to define ports by similar functionality or role. You can create port groups and use them to configure QoS profiles, ingress meters and flood rate-limiters for ports that share the same role. QoS commands that use port groups are updated

automatically if the ports group is removed or if ports are added or removed from the group.

In addition, ExtremeCloud IQ Site Engine can create and manipulate port groups through the CoS MIB using SNMP. These groups are used to configure the ExtremeCloud IQ Site Engine-specified CoS resources that are mapped to QoS profiles, ingress meters, and flood rate-limiters. Since the ExtremeCloud IQ Site Engine CoS feature is port group-oriented, ExtremeXOS QoS entities (QoS profiles, ingress meter, and flood rate-limiting) that are configured on a per-port basis are not exposed through the CoS MIB using SNMP.

To create port groups, use the following command:

```
create ports group port_group
```

To delete port groups, use the following command:

```
delete ports group port_group
```

To add/delete ports from a port group, use the following command:

```
configure ports {group} port_group [[ add | delete ] port_list ]
```

To show port group information, use the following command:

```
show ports group {port_group}
```

CoS Port Resource

The TXQ, IRL and Flood Control port resource tables in *CoS* are configured using the *QoS* profile, ingress meter and rate-limit flood resource commands in the CLI. These port resources are mapped to CoS when the resources in the CLI are configured using a port group.

CoS Global Enable Action

An *SNMP* set to enable *CoS* will automatically enable per-port dot1p examination and diffserv replacement controls as needed for the functionality of CoS. Diffserv replacement is enabled for a *QoS* profile on all ports if the corresponding tosValue for CoS index 0-7 is configured. Otherwise, diffserv replacement is disabled for the corresponding QoS profile on all ports. Dot1p replacement is disabled for all QoS profiles on all ports. In addition, all QoS profiles are created and the dot1p examination to QoS profile mapping are fixed to match the TXQ reference mapping for CoS indexes 0-7. The internal CoS state is used to warn the user in the CLI in the event that the controls setup by CoS are modified. A subsequent SNMP set to disable CoS disables the internal CoS state. This turns off the warning messages in the CLI; however, it does not restore the configuration to its original state before the global enable action occurred.

When the internal CoS state is enabled, the global enable action is performed on any new switch that is inserted into the stack.

Meter and Flood Actions

ExtremeXOS 16.1 has been expanded to implement the syslog, trap, and disable-port out-actions for meters and flood rate-limiters. This requires additional counters and a polling task to periodically check the thresholds for the configured events.

Meter and Flood Limitations

Port-Meter Out-of-profile Limitations

The *CoS* MIB supports an out-of-profile status and counter for each per-port meter on each port. On all 16.1 supported platforms, the hardware only supports a global counter for each per-port meter. The out-of-profile status is supported for each per-port meter on each port; however, the status is only valid while the meter is out-of-profile. When the per-port meter rate is no longer exceeded, the out-of-profile status is automatically cleared in the hardware. The ability to detect an out-of-profile per-port meter on a given port is dependent on the out-of-profile status of the per-port meter at the time of the software polling interval.

Due to this limitation, the out-of-profile actions can only accurately be performed if the exceeded per-port meter rate is sustained for the duration of the software polling interval. Any change in the global out-of-profile counter for the per-port meter will be evenly distributed between the ports that indicate an out-of-profile status at the time of the software polling interval. This should give a rough approximation of the out-of-profile count of the per-port meter on each port.

Flood Out-of-profile Limitations

The CoS MIB supports an out-of-profile status and counter on each port for each flood type (unknown unicast, multicast, and broadcast). On all ExtremeXOS 16.1 supported platforms, the hardware only supports a single counter and status per port for all flood types. Any out-of-profile actions configured for one flood type on a given port will be performed if any flood type is out-of-profile on that port. The out-of-profile counter will be the aggregate count of all flood types on a given port.

Configuring QoS

Configuring QoS on SummitStack and ExtremeSwitching Switches

Figure 184 shows the *QoS* configuration components for SummitStack and ExtremeSwitching switches.

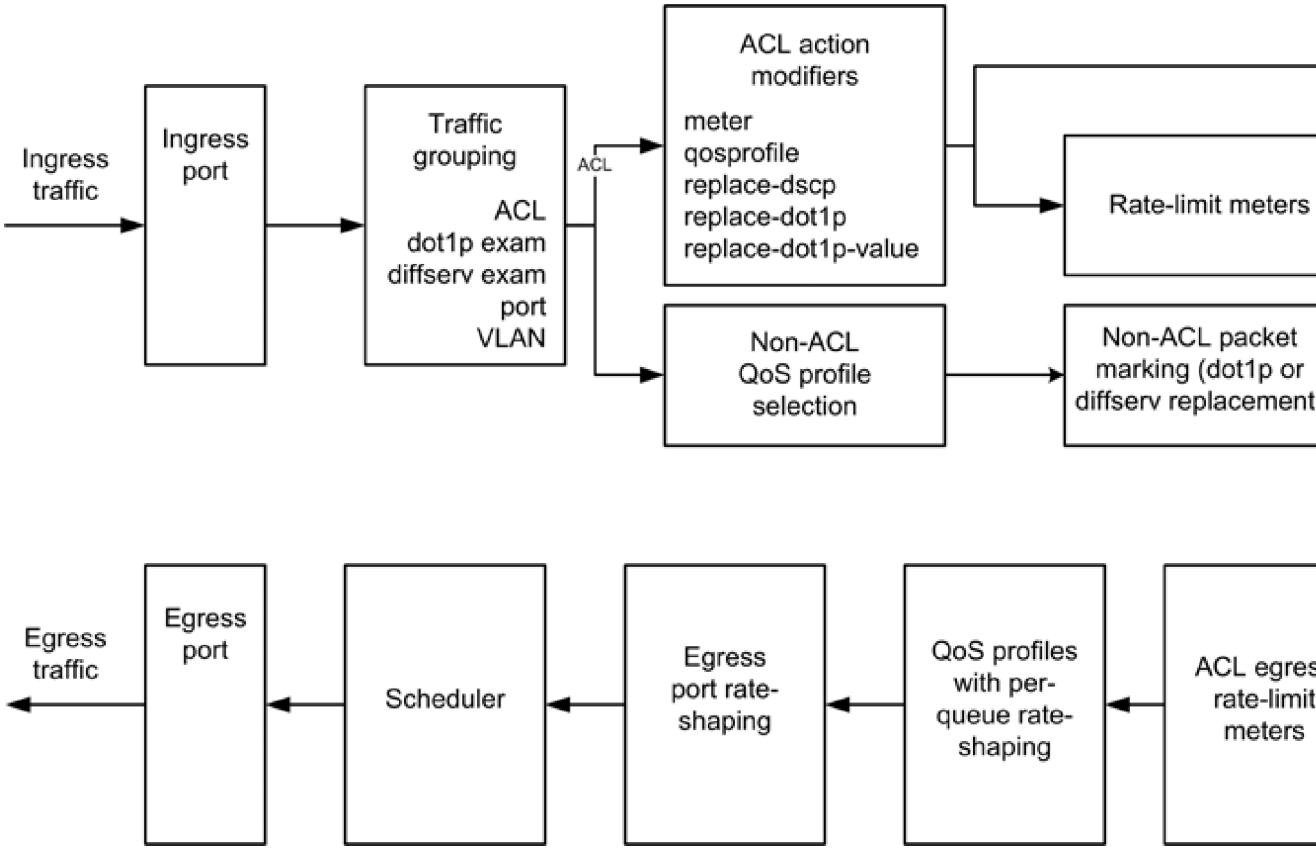


Figure 184: QoS on SummitStack and ExtremeSwitching Switches

The considerations below apply to QoS on SummitStack and ExtremeSwitching switches:

- The following QoS features share resources:
  - *ACL*s
  - dot1p
  - *VLAN*-based QoS
  - Port-based QoS
- You might receive an error message when configuring a QoS feature in the above list; it is possible that the shared resource is depleted. In this case, unconfigure one of the other QoS features and reconfigure the one you are working on.
- On a SummitStack, you cannot create QoS profile QP7. This QoS profile is reserved for system control traffic.
- These switches allow dynamic creation and deletion of QoS queues, with QP1 and QP8 always available.
- ACL egress rate-limit meters are supported only on all platforms.

Configuration Summary

Use the following procedure to configure *QoS*:

1. Configure basic Layer 2 connectivity (prerequisite).

2. Configure QoS scheduling, if needed, as described in Selecting the QoS Scheduling Method on page 1477.
3. Configure ingress and egress rate-limiting as needed:
   a. Create a meter as described in Creating Meters on page 1485.
   b. Configure the meter as described in Configuring a Meter on page 1485.
   c. Apply the meter to ingress traffic as described in Applying a Meter to Ingress Traffic on page 1486.
4. Configure non-*ACL*-based egress QoS profile selection as described in the following sections:
   - Configuring a CoS 802.1p-Based Traffic Group on page 1482
   - Configuring a DiffServ-Based Traffic Group on page 1483
   - Configuring a Port-Based Traffic Group on page 1484
   - Configuring a VLAN-Based Traffic Group on page 1485
5. Configure 802.1p or DiffServ packet marking as described in Configuring 802.1p or DSCP Replacement on page 1477.
6. Configure egress QoS profile rate shaping as needed:
   a. Create egress QoS profiles as described in Creating or Deleting an Egress QoS Profile on page 1480.
   b. Configure egress QoS profile rate shaping parameters as described in Configuring an Egress QoS Profile on page 1480.
7. Configure egress port rate shaping as described in Configuring Egress Port Rate Limits on page 1481.
8. Finalize ACL traffic-based group configuration as described in Configuring an ACL-Based Traffic Group on page 1482.
9. Verify the configuration using the commands described in Displaying QoS Configuration and Performance on page 1487.

Quality of Service Port Speed Calculation

Version 33.3.1 adds an enhancement to the Quality of Service (QoS) profile that lets you use the "minbw" and "maxbw" settings to calculate the bandwidth based on the actual port speed and not on the maximum port speed.

Supported Platforms

All platforms.

Modified CLI Commands

The `configure qosprofile` command adds the **`minbw_actual`** *`minbw_actnum`* and **`maxbw_actual`** *`maxbw_actnum`* options.

The `show configuration`, `show port information`, and `show qosprofile` commands add **`minbw_actual`** and **`maxbw_actual`** information.

Selecting the QoS Scheduling Method

QoS scheduling determines the order of QoS profile service and varies between platforms. SummitStack and ExtremeSwitching series switches support three scheduling methods: strict-priority, weighted-round-robin, and weighted-deficit-round-robin. These scheduling methods are described in Scheduling on page 1464. Scheduling can be applied globally or on a per-port or per-PortGroup basis. If no port_list or port_group is specified in the command, the global scheduling algorithm is set and applied to ports that are not configured via a port_list or port_group. Specifying a port_list or port_group in the command configures the scheduling algorithm for specific ports.

QoS profiles can also be overriden on a global, per-port or per-port_group basis. If no port_list or port_group is specified, the global qosprofile configuration is changed and applied to all ports that are not configured on a per-port or per-port_group basis.

1. Select the QoS scheduling method for a switch.

   ```
   configure qosscheduler [strict-priority | weighted-round-robin |
   weighted-deficit-round-robin ] {ports [port_list | port_group | all ]}
   ```

   Note
   In a SummitStack, the scheduling algorithm is automatically programmed by ExtremeXOS for the stacking links only, and will likely be different than the algorithm you select.

2. Override the weighted-round-robin switch configuration on a specific QoS profile.

   ```
   configure qosprofile qosprofile use-strict-priority
   ```

Configuring 802.1p or DSCP Replacement

Replacement of 802.1p Priority Information on Egress

By default, 802.1p priority information is not replaced or manipulated, and the information observed on ingress is preserved when forwarding the frame. This behavior is not affected by the switching or routing configuration of the switch. However, the switch is capable of inserting and/or overwriting 802.1p priority information when it transmits an 802.1Q tagged frame as described below.

Replacement in ACL-Based Traffic Groups

If you are using ACL-based traffic groups, you can use the `replace-dot1p` action modifier to replace the ingress 802.1p priority value with the 802.1p priority value of the egress *QoS* profile as listed in the following table. To specify a specific 802.1p priority value on egress, use the `replace-dot1p-value` action modifier.

Note
If you are using ACL-based traffic groups, you must use ACL action modifiers to replace the 802.1p priority. Traffic that meets any ACL match conditions is not subject to non-ACL-based 802.1p priority replacement.

Replacement in Non-ACL-Based Traffic Groups

For non-ACL-based traffic groups, you can enable or disable 802.1p priority replacement on specific ingress ports. When 802.1p priority replacement is enabled, the default egress 802.1p priority value is set to the priority value of the egress QoS profile as listed in Table 132.

Table 132: Default Queue-to-802.1p Priority Replacement Value

Egress QoS Profile	802.1p Priority Replacement Value
QP1	0
QP2	1
QP3	2
QP4	3
QP5	4
QP6	5
QP7	6
QP8	7

To enable 802.1p priority replacement on egress, use the following command:

```
enable dot1p replacement ports [port_list | all]
```

Note
The port in this command is the ingress port.

To disable this feature, use the following command:

```
disable dot1p replacement ports [port_list | all]
```

Note
If only DiffServ traffic groups are enabled, then 802.1p priority enforcement for 802.1q tagged packets continues for non-IP packets using the default 802.1p map shown in the following table.

Only QP1 and QP8 exist by default; you must create QP2 to QP7 (QP2 to QP5 in a SummitStack). If you have not created these QPs, the replacement feature will not take effect.

When DiffServ examination is enabled on 1 Gigabit Ethernet ports, 802.1p replacement is enabled and cannot be disabled. The ingress 802.1p value is replaced with the 802.1p value assigned to the egress QoS profile.

Replacement of DSCP on Egress

The switch can be configured to change the DSCP in a packet before forwarding the packet. This is done with no impact on switch performance and can be configured as described in the following sections:

[11] You cannot direct traffic to this QoS profile in a SummitStack.

Replacement in ACL-Based Traffic Groups

If you are using ACL-based traffic groups, you can use the replace-dscp action modifier to replace the ingress DSCP value with the DSCP value of the egress *QoS* profile as listed in Table 133. This action modifier functions for both IPv4 and IPv6 traffic.

Note
If you are using ACL-based traffic groups, you must use ACL action modifiers to replace the DSCP. Traffic that meets any ACL match conditions is not subject to non-ACL-based DSCP priority replacement. For all platforms, we recommend that you use ACL-based traffic groups when configuring DSCP replacement.

Replacement in Non-ACL-Based Traffic Groups

For non-ACL-based traffic groups, you can enable or disable DSCP replacement on specific ingress ports. When DSCP replacement is enabled, the DSCP value used on egress is determined by either the QoS profile or the 802.1p priority value. Table 133 shows the default mappings of QoS profiles and 802.1p priority values to DSCPs.

Table 133: Default QoS Profile and 802.1p Priority Value Mapping to DiffServ Code Points

QoS Profile	802.1p Priority Value	DSCP
QP1	0	0
	1	8
	2	16
	3	24
	4	32
	5	40
	6	48
QP8	7	56

Replacing a DSCP on Egress

- To replace DSCPs by enabling DiffServ replacement, use the command:
  `enable diffserv replacement ports [port_list | all]`

  Note
  The port in this command is the ingress port.

- To disable this feature, use the command:
  `disable diffserv replacement ports [port_list | all]`
- To view the current DiffServ replacement configuration, use the command:
  `show diffserv replacement`
- To change the DSCP mapping, use the commands:
  `configure diffserv replacement [{qosprofile} qosprofile | priority priority] code-point code_point`
  `unconfigure diffserv replacement`

DiffServ Example

In this example, we use DiffServ to signal a class of service throughput and assign any traffic coming from network 10.1.2.x with a specific DSCP. This allows all other network switches to send and observe the DSCP instead of repeating the same *QoS* configuration on every network switch.

To configure the switch:

1. Using ACLs, assign a traffic grouping for traffic from network 10.1.2.x to QP3:

   ```
   configure access-list qp3sub any
   ```

   The following is a sample policy file example:

   ```
   #filename: qp3sub.pol
   entry QP3-subnet {
       if {
           source-address 10.1.2.0/24
        } then {
           Qosprofile qp3;
           replace-dscp;
           }
   ```

2. Configure the switch so that other switches can signal calls of service that this switch should observe.

   ```
   enable diffserv examination ports all
   ```

   Note
   The switch only observes the DSCPs if the traffic does not match the configured access list. Otherwise, the *ACL* QoS setting overrides the QoS DiffServ configuration.

Configuring Egress QoS Profile Rate Shaping

Creating or Deleting an Egress QoS Profile

The default configuration for most platforms defines eight egress *QoS* profiles. The default configuration defines two egress QoS profiles.

- Use the following command to create an additional egress QoS profile:

  `create qosprofile [QP2| QP3 | QP4 | QP5 | QP6 | QP7]`

- Use the following command to delete an egress QoS profile:

  `delete qosprofile [QP2| QP3 | QP4 | QP5 | QP6 | QP7]`

Configuring an Egress QoS Profile

- Egress *QoS* profile rate shaping is disabled by default on all ports. On all platforms, use the following commands to configure egress QoS profile rate shaping on one or more ports:

  `configure qosprofile` **`egress`** *`qosprofile`* `[{`**`minbw`** *`minbw_number`*`} {`**`maxbw`** *`maxbw_number`*`} | {`**`peak_rate`** *`peak_bps`* `[`**`K`** `|` **`M`**`]}] [`**`ports`** `[`*`port_list`* `|` **`all`**`]]`

```
configure qosprofile qosprofile [{minbw minbw_number} {maxbw
maxbw_number} | {{committed_rate committed_bps [K | M]} {peak_rate
peak_bps [K | M]} | [ports [port_list | all]]
configure {qosprofile} qosprofile [[{maxbuffer buffer_percentage}
{use-strict-priority}] | [maxbuffer buffer_percentage ports [port_list
| all]]]
```

Note

You must use these commands on all platforms if you want to configure the buffer size or weight value. Otherwise, you can use the command in the following description.

You cannot configure the priority for the QoS profile.

- To remove the limit on egress bandwidth per QoS profile per port, re-issue this command using the default values.
- To display the current configuration for the QoS profile, use the following command.
  ```
  show qosprofile [ all | port_list | port_group]
  ```

Configuring WRED on an Egress QoS Profile

- To configure or unconfigure WRED on an egress *QoS* profile, use the following commands:
  ```
  configure {qosprofile} {egress} qosprofile [wred [{color [tcp [green |
  red] | non-tcp [any|red]] [{min-threshold min_thresh} {max-threshold }
  {max-drop-rate max_drop_rate}]} | avg-weight avg_weight]] ports
  [port_list | all]
  unconfigure qosprofile wred {ports [port_list | all]}
  ```
- To display the WRED configuration settings, use the following command:
  ```
  show wredprofile {ports [port_list | all]}
  ```

Configuring Egress Port Rate Limits

Configuring Egress Traffic

- The default behavior is to have no limit on the egress traffic per port. To configure egress rate limiting, use the following command:
  ```
  configure ports [port_list | port_group]rate-limit egress [no-limit |
  cir-rate [Kbps | Mbps | Gbps] {max-burst-size burst-size [Kb | Mb]}]
  ```
- To view the configured egress port rate-limiting behavior, use the following command.
  ```
  show port {mgmt | port_list | tag tag} information {detail}
  ```

  You must use the *detail* parameter to display the egress port rate configuration and, if configured, the maximum burst size. Refer to Displaying Port Information on page 331 for more information on the `show ports information` command.

  You can also display this information using the following command:
  ```
  show configuration vlan
  ```

The following is sample output from the show configuration vlan command for configured egress rate limiting:

```
# Module vlan configuration.
#
configure vlan Default tag 1
config port 3:1 rate-limit egress 128 Kbps max-burst-size 200 Kb
config port 3:2 rate-limit egress 128 Kbps
config port 3:10 rate-limit egress 73 Kbps max-burst-size 128 Kb
configure vlan Default add ports 3:1-48 untagged
```

Note
Refer to FDB for more information on limiting broadcast, multicast, or unknown MAC traffic ingressing the port.

Configuring Traffic Groups

Configuring an ACL-Based Traffic Group

ACL-based traffic groups are introduced in ACL-Based Traffic Groups on page 1457. An ACL can implement multiple *QoS* features, so it is usually best to finalize the ACL after all other features have been configured.

To configure an ACL-based traffic group, do the following:

1. Create an ACL policy file and add rules to the file using the following guidelines:
   a. Use ACL match conditions to identify the traffic for the traffic group.
   b. Use ACL action modifiers to apply QoS features such as ingress meter or traffic queue selection, egress QoS profile or traffic queue selection, and 802.1p priority replacement to the traffic group.
2. Apply the ACL policy file to the ports where you want to define the traffic groups. You can apply the file to specific ports, all ports, or all ports in a *VLAN*.

Note
ACLs are described in detail in the ACLs chapter.

Configuring a CoS 802.1p-Based Traffic Group

As described in CoS 802.1p-Based Traffic Groups on page 1458, the default switch configuration defines *CoS* 802.1p-based traffic groups. The configuration options for these groups are described in the following sections:

- Enabling and Disabling 802.1p Examination on page 1483
- Changing the 802.1p Priority to QoS Profile Mapping on page 1483

Note
If you are using *ACL*-based traffic groups, use the `qosprofile` or `traffic-queue` action modifier to select a forwarding queue. Traffic that meets any ACL match conditions is not evaluated by other traffic groups.

Enabling and Disabling 802.1p Examination

CoS 802.1p examination is supported on all platforms and enabled by default. However, you can only disable and enable this feature. To free *ACL* resources, disable this feature whenever another *QoS* traffic grouping is configured. (See ACLs for information on available ACL resources.)

Note
If you disable this feature when no other QoS traffic grouping is in effect, 802.1p priority enforcement of 802.1q tagged packets continues. If only DiffServ traffic groups are enabled, then 802.1p priority enforcement for 802.1q tagged packets continues for non-IP packets using the default 802.1p map shown in the following table.

- To disable the 802.1p examination feature, use the following command.

  ```
  disable dot1p examination ports [port_list | all]
  ```

  Note
  802.1p examination cannot be disabled for 802.1p priority level 6 in a SummitStack. When 802.1p examination is disabled on a SummitStack, the precedence for the remaining examination becomes lower than that of all other traffic groupings.

- To re-enable the 802.1p examination feature, use the following command:

  ```
  enable dot1p examination ports [port_list | all]
  ```

- To display whether the 802.1p examination feature is enabled or disabled, use the following command:

  ```
  show ports {mgmt | port_list | tag tag} information {detail}
  ```

Changing the 802.1p Priority to QoS Profile Mapping

To view the current 802.1p priority to *QoS* profile mapping on a switch, enter the following command:

```
show dot1p
```

To change the mapping, enter the following command:

```
configure dot1p type dot1p_priority {qosprofile} qosprofile
```

Configuring a DiffServ-Based Traffic Group

As described in DiffServ-Based Traffic Groups on page 1459, the default switch configuration defines DiffServ-based traffic groups. The configuration options for these groups are described in the following sections:

- Enabling and Disabling Diffserv Examination on page 1484

- Changing the DSCP to QoS Profile Mapping on page 1484

Note
If you are using *ACL*-based traffic groups, use the `qosprofile` or `traffic-queue` action modifier to select a forwarding queue. Traffic that meets any ACL match conditions is not evaluated by other traffic groups.

Enabling and Disabling Diffserv Examination

When a packet arrives at the switch on an ingress port and Diffserv examination is enabled, the switch uses the DSCP value to select the egress *QoS* profile that forwards the packet. The QoS profile configuration defines the forwarding characteristics for all traffic assigned to the QoS profile.

- The DiffServ examination feature is disabled by default. To enable DiffServ examination, use the following command:

  `enable diffserv examination ports [port_list | all]`

  Note
  When DiffServ examination is enabled on 1 Gigabit Ethernet ports for ExtremeSwitching series switches, 802.1p replacement is enabled and cannot be disabled. The ingress 802.1p value is replaced with the 802.1p value assigned to the egress QoS profile.

- To disable DiffServ examination, use the following command:

  `disable diffserv examination ports [port_list | all]`

Changing the DSCP to QoS Profile Mapping

- You can change the egress *QoS* profile assignment for each of the 64 code points. To view the current DSCP to QoS profile mapping, use the following command:

  `show diffserv examination`

- Use the following commands to change the DSCP to QoS profile mapping:

  `configure diffserv examination code-point code_point {qosprofile} qosprofile`

  `unconfigure diffserv examination`

  After a QoS profile is assigned, the rest of the switches in the network prioritize the packet using the characteristics specified by the QoS profile.

Configuring a Port-Based Traffic Group

A port-based traffic group links a physical ingress port to an egress *QoS* profile for traffic forwarding.

To configure a port-based traffic group, use the following command:

```
configure ports port_list {qosprofile} qosprofile
```

Note

If you are using *ACL*-based traffic groups, use the `qosprofile` or `traffic-queue` action modifier to select a forwarding queue. Traffic that meets any ACL match conditions is not evaluated by other traffic groups.

Port-based traffic groups apply to all packets.

Configuring a VLAN-Based Traffic Group

A *VLAN*-based traffic group links all ports in a VLAN to an egress *QoS* profile for traffic forwarding. All intra-VLAN switched traffic and all routed traffic sourced from the named VLAN uses the specified QoS profile.

To configure a VLAN-based traffic group, use the following command:

```
configure vlan vlan_name {qosprofile} qosprofile
```

Note

If you are using *ACL*-based traffic groups, use the `qosprofile` or `traffic-queue` action modifier to select a forwarding queue. Traffic that meets any ACL match conditions is not evaluated by other traffic groups.

VLAN-based traffic groups apply to all packets.

Creating and Managing Meters

You can configure meters to define bandwidth requirements on SummitStack and ExtremeSwitching series switches.

Creating Meters

- To create a meter, use the following command:

  ```
  create meter meter-name
  ```

- To display the meters already configured on the switch, use the command:

  ```
  show meter meter_name
  ```

ExtremeXOS now creates 8 to 16 default ingress meters that are used for per-port metering. These are named "ingmeter<n>" where n is 0-15.

Configuring a Meter

After you create a meter, you can configure the meter using the command for the platform you are using. To configure a *QoS* meter on all platforms, use the following command:

```
configure meter metername {committed-rate cir [Gbps | Mbps | Kbps |
Pps]} {max-burst-size burst-size [Kb | Mb | packets]} {out-actions
[{disable-port} {drop | set-drop-precedence {dscp [none | dscp-value]}
```

```
{dot1p [ none | dot1p-value ]}} {log} {trap}]} {ports [port_group |
port_list]}
```

Note
The disable-port, log, and trap options are only allowed for per-port meters (i.e. ingmeter0, ingmeter1,...).

Applying a Meter to Ingress Traffic

You can apply a meter to ingress traffic using an *ACL* on SummitStack and ExtremeXOS switches.

Use rules within the ACL to identify the ingress traffic to which you want to apply the meter. Apply the meter by specifying the meter name with the `meter metername` action modifier. For information on completing the ACL configuration, see Configuring an ACL-Based Traffic Group on page 1482.

Deleting a Meter

To delete a meter, use the following command:

```
delete meter meter-name
```

Note
The associated meters are not deleted when you delete any type of traffic queue. Those meters remain and can be associated with other traffic queues. To display the configured meters, use the `show meter meter_name` command.

Adjusting the Byte Count Used to Calculate Traffic Rates

You can configure a per-packet byte adjustment that the system uses to calculate the ingress traffic rate, traffic utilization, and traffic statistics. You configure either the number of bytes you want subtracted from each packet ingressing the specified ports or the number of bytes you want added to the packet ingressing the specified ports.

By default, all bytes are counted for the ingressing traffic rate. After you issue this command, the default number of bytes removed is 0; you can add or subtract from one to four bytes from each ingressing packet on the specified ports for calculating the ingressing traffic rate.

- To display the number of bytes added to or subtracted from the packet to calculate the ingressing traffic rate, traffic utilization, and traffic statistics, enter the command:

  ```
  show ports port_list information detail
  ```

  Note
  You must use the detail keyword to display this information.

- To unconfigure this setting, re-issue the command and enter the value `0`.

Controlling Flooding, Multicast, and Broadcast Traffic on Ingress Ports

You can control ingress flooding of broadcast and multicast traffic and traffic for unknown destination MAC addresses.

To control ingress flooding of broadcast and multicast traffic and traffic for unknown destination MAC addresses, enter the command:

```
configure ports port_list | port_group rate-limit flood [broadcast | multicast | unknown-destmac] [no-limit | pps]
```

Displaying QoS Configuration and Performance

Displaying Traffic Group Configuration Data

Displaying 802.1p Priority to QoS Profile Mappings

To display the 802.1p priority to egress *QoS* profile mappings, enter the command:

```
show dot1p
```

Displaying DiffServ DSCP to QoS Profile Mappings

To display the DiffServ DSCP to egress *QoS* profile mappings, enter the command:

```
show diffserv examination
```

Displaying Port and VLAN QoS Settings

To display *QoS* settings assigned to ports or *VLAN*s, enter the command:

```
show port {mgmt | port_list | tag tag} information {detail}
```

Note
To ensure that you display the QoS information, you must use the detail keyword.

This command displays which QoS profile, if any, is configured.

Displaying Performance Statistics

Displaying QoS Profile Traffic Statistics

After you have created *QoS* policies that manage the traffic through the switch, you can use the QoS monitor to determine whether the application performance meets your expectations.

The QoS monitor allows you to display the traffic packet counts in a real-time or a snapshot display for the specified ports.

- View QoS profile traffic statistics entering the command:.

  ```
  show ports port_list qosmonitor {congestion} {no-refresh}
  ```

- View or clear the WRED statistics by entering the command:

  ```
  show ports port_list wred {no-refresh}
  clear counters wred
  ```

Displaying Congestion Statistics

- To display a count of the packets dropped due to congestion on a port, use the command:

  ```
  show ports port_list congestion {no-refresh}
  ```

- To display a count of the packets dropped due to congestion in the *QoS* profiles for a port, use the command:

  ```
  show ports port_list qosmonitor {congestion} {no-refresh}
  ```

- To display traffic counts for each profile on the specified stacking ports, use the command:

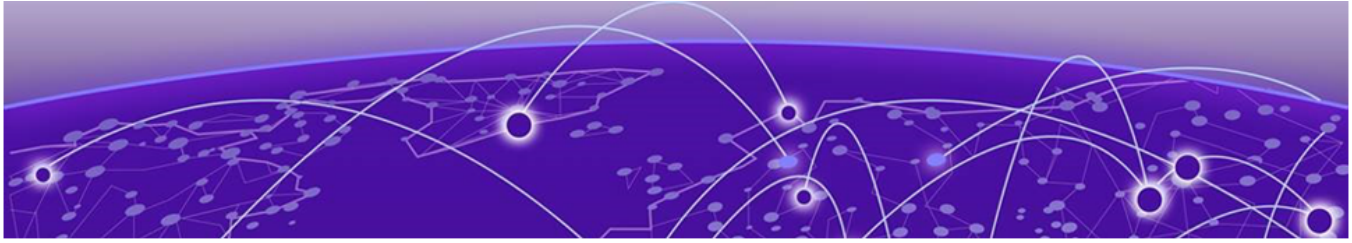
  ```
  show ports stack-ports stack_port_list qosmonitor {no-refresh | refresh} {port-number}
  ```

- To display packet counts or dropped-traffic counts for each profile on the specified stacking ports, use the command:

  ```
  show ports stack-ports stack_port_list qosmonitor congestion {no-refresh | refresh} {port-number}
  ```

- To display the stacking port congestion statistics (dropped packets) for the specified ports on the front panel, use the command:

  ```
  show ports stack-ports stack_port_list congestion {no-refresh | refresh} port-number
  ```



RADIUS over TLS

TLS Connection Persistence on page 1489
PKI Authentication of TLS Connections on page 1490
Revocation Checking of Server Certificates via OCSP on page 1490
Limitations on page 1491
Supported Platforms on page 1491
RADIUS Over TLS CLI Commands on page 1492
Setting Up FreeRADIUS TLS on page 1492
RADIUS TLS Cipher Configuration on page 1494
Reference Identifier Support for RADIUS TLS on page 1494

This chapter introduces Remote Authentication Dial In User Service (RADIUS) over Transport Layer Security (TLS), an option for transporting RADIUS packets over the secure, reliable, and connection-oriented TLS protocol as defined by RFC 6614.

Using RADIUS over TLS enables dynamic trust relationships between RADIUS servers. Up to eight (8) RADIUS servers are configurable. If all servers are configured as User Datagram Protocol (UDP), then a round-robin algorithm is used to determine which UDP server the initial transmission will be sent to. If more than one TLS servers is live, the highest priority TLS server is chosen for the first transmission.

When a RADIUS server is configured for TLS:

- The destination port is 2083/TCP (radsec).
- The RADIUS shared secret is radsec.
- The TLS connection is authenticated using X.509 certificates.

When a mixture of UDP and TLS servers are configured, TLS takes priority. The only time the UDP servers will be used is if all TLS servers are down.

TLS Connection Persistence

With RADIUS over TLS, the state of the TLS connection indicates whether the server is “live” or “dead.” As soon as a RADIUS over TLS server is configured, the switch attempts to open a connection. If successful, the server is considered live. If unsuccessful, the server is considered dead and the switch will periodically attempt to reconnect.

If one or more RADIUS over TLS servers is live, then an initial request is sent to a single TLS server, based on priority. This means that the highest priority TLS server is always chosen for the first transmission. If this transaction times out, but that TLS server status is live, there will be no retransmissions to other TLS servers (or to UDP servers, if configured) and user authentication will fail.

When a TLS connection is broken or closed, any prior request messages that have not yet received a response will be handled on timeout as follows:

- If there are no other live TLS connections, then the user session will fail. If there are UDP servers configured, they will not be tried.
- If there are other live TLS connections, then the user request will be retransmitted to one of the new TLS servers.

PKI Authentication of TLS Connections

Each TLS connection is authenticated using existing, customer-provisioned SSL X.509 certificates and private keys. The RADIUS over TLS connection is established by the AAA application, passing a pointer to these credentials to the OpenSSL cryptographic library.

If there are no live RADIUS over TLS connections, then the AAA application will fall back on legacy RADIUS over UDP (if configured).

Failed RADIUS over TLS connections are retried once every minute, indefinitely. If any live server connections go down, ExtremeXOS will automatically make a reconnect attempt every five seconds for one minute. After one minute, the reconnect attempt occurs every five minutes.

Revocation Checking of Server Certificates via OCSP

In addition to checking the server certificate's validity (signatures, expiration date, uses), the switch also checks the revocation status of each certificate in the chain using the Online Certificate Status Protocol (OCSP).

The following rules are enforced:

- The location of the OSCP server must be embedded in the certificate being verified. If missing, the certificate will not be trusted.
- If the OCSP server is not reachable, the certificate will not be trusted.
- If the OSCP server reports that the certificate has been revoked, the certificate will not be trusted.
- Every certificate in the chain will be revocation checked (except for the Root certificate, which is not revokable by definition).
- The OCSP response must be signed. The switch supports all three OCSP trust models:
  - Common Issuer: Certificate Authority (CA) that signs cert also signs OSCP response
  - Trusted Responder Model (TRM): OCSP response signed by a self-signed certificate that is trusted by the switch for this purpose

- Delegated Trust Model (DTM): CA that signs cert issues the CA that signs OCSP response

Beginning with ExtremeXOS 31.6, you can configure RADIUS over TLS OCSP attributes (nonce, override, and ocsp-nocheck, respectively) using the following commands:

- `configure radius` **`tls ocsp nonce`** `[`**`on`** | **`off`**`]`
- `configure radius` **`tls ocsp override`** `[`*`url`* | **`none`**`]`
- `configure radius` **`tls ocsp signer ocsp-nocheck`** `[`**`on`** | **`off`**`]`

OCSP nonce cryptographically binds an OCSP request and an OCSP response with an `id-pkix-ocsp-nonce` extension to prevent replay attacks.

OCSP override configures one HTTP Online Certificate Status Protocol (OSCP) override URL for a RADIUS TLS server.

When OCSP-nocheck is done for a peer certificate, ExtremeXOS sends the OCSP request to the OCSP server. The OCSP response is signed by the OCSP responder/signer. The response also comes along with the certificate of the OCSP signer. When ExtremeXOS receives the response, it only checks whether the status of the peer certificate is not revoked.

Limitations

The following features of Change-of-Authorization (RFC5176) are not implemented in ExtremeXOS:

- Reverse Path Forwarding Check—Typically this is used in a proxy scenario. This check is used to determine if the IP address indicated by the *RADIUS (Remote Authentication Dial In User Service)* attributes is a routable destination address for a request sent by the switch software.
- IPSEC encryption—End-to-end encryption of both the RADIUS requests and responses.
- Disconnect-Request and Change-of-Authorization packets identifying sessions with anything other than the Calling-Station-Id attribute containing a properly formatted MAC address. In addition to the Calling-Station-ID attribute, you can also use a NAS-Port attribute, which indicates the index of the specific port the session is connected to.
- Acct-Session-Id attribute—This is an alternate means of session identification. Sessions are currently uniquely identified by port and MAC address pair.
- Retransmissions of Disconnect-Request or Change-of-Authorization ACK and NAK packets—Retransmissions of packets is the responsibility of the device initiating the dynamic authorization transactions.

Supported Platforms

MSDP (Multicast Source Discovery Protocol) is supported on all platforms running a minimum software version of ExtremeXOS 12.0 with the Premier or Extreme Platform ONE license.

Our implementation of MSDP is compliant with RFC 3618 and RFC 3446, and compatible with other devices that are compliant with these standards.

RADIUS Over TLS CLI Commands

The following commands support RADIUS over TLS:

```
configure radius {mgmt-access | netlogin} [primary | secondary |
index] server [host_ipaddr | host_ipV6addr | hostname] {udp_port | tls
{tls_port}} client-ip [client_ipaddr | client_ipV6addr] {vr vr_name}
{shared-secret {encrypted} secret
```

```
configure radius-accounting { mgmt-access | netlogin } [ primary |
secondary | index ] server [ host_ipaddr | host_ipV6addr | hostname]
{udp_port | tls {tls_port}} client-ip [ client_ipaddr | client_ipV6addr]
{vr vr_name} {shared-secret {encrypted} secret}
```

```
configure radius tls ocsp nonce [on | off]
```

```
configure radius tls ocsp [ on | off]
```

```
configure radius tls ocsp override [url | none]
```

```
configure radius tls ocsp signer ocsp-nocheck [on | off]
```

```
configure radius tls tcp-user-timeout [ seconds | default]
```

Setting Up FreeRADIUS TLS

The following is an example setup configuration of FreeRADIUS as a RADIUS TLS server.

1. Edit the following CA config file (`ca.cnf`) to have key usage:

```
/etc/freeradius/3.0/certs   (otherwise /etc/raddb/certs)

    [v3_ca]
    subjectKeyIdentifier    = hash
    authorityKeyIdentifier  = keyid:always,issuer:always
    basicConstraints        = critical,CA:true
    #crlDistributionPoints  = URI:http://www.example.org/example_ca.crl
    keyUsage = cRLSign, keyCertSign, digitalSignature
    extendedKeyUsage = OCSP Signing
    noCheck = yes
```

Note
ExtremeXOS requires key usage for a CA certificate.

2. Edit the file `/etc/freeradius/3.0/certs/xpextensions` with the OCSP server address:

```
[ xpserver_ext] extendedKeyUsage = 1.3.6.1.5.5.7.3.1
    authorityKeyIdentifier = keyid,issuer
    authorityInfoAccess = OCSP;URI:http://10.127.2.236:2561
```

Note
Use the IP of the machine in which the OCSP server runs. It can be the same machine where FreeRADIUS runs. DNS can also be used instead of the IP.

3. Run the OCSP server using the following command in the same path where the certificates and `index.txt` are present:

```
openssl ocsp -port 2561 -text -index index.txt -CA ca.pem -rkey ca.key -rsigner ca.pem
```

4. Clean up by entering **`make destroycerts`**.
5. Create the RADIUS TLS server key and certificate by entering **`make server.pem`**
6. Create the ExtremeXOS switch key and certificate by entering **`make client.pem`**.
7. Create the DH by entering **`make dh`**.
8. Enable TLS in FreeRADIUS:

   `File: etc/freeradius/3.0/sites-available/tls`

   a. If you do not want the RADIUS server authenticating ExtremeXOS, search for the previous command line and set **`require_client_cert= no`**.

   b. Search for 'clients radsec' and add the ExtremeXOS switch IP:

```
clients radsec {
                client 10.127.2.19 {
                        ipaddr = 10.127.2.19
                        proto = tls
                        secret = radsec
                }
```

   c. Add a soft link for etc/freeradius/3.0/sites-available/tls at etc/freeradius/3.0/sites-available/tls by running **`ln -s <file>`**.

   d. Restart FreeRADIUS (or radiusd).

```
EXOS Switch Setup:
------------------
```

9. Copy the `ca.pem` to the TFTP server.
10. Copy the `client.pem` to the TFTP server.
11. Convert the private, encrypted client key to plain format:

```
openssl rsa -in client.key -out clientPlain.key
```

12. Copy the `clientPlain.key` to the TFTP server.
13. Install the following to the switch:

    a. Download the `ca.pem` as a trusted ca in the switch by entering **`download ssl <ip> certificate trusted-ca <file>`**

    b. Download the `client.pem` as an ssl-cert in the switch by entering **`download ssl <ip> certificate ssl-cert <file>`**

    c. Download the `clientPlain.key` as priv-key in the switch by entering **`download ssl <ip> privkey <file>`**

14. Configure RADIUS or RADIUS-accounting with the TLS option.

RADIUS TLS Cipher Configuration

Version 33.3.1 adds support for RADIUS TLS cipher configuration. Additionally, for ECDHA ciphers, the curves list is fixes as secp256r1, secp384r1, and secp521r1. The following ciphers are supported:

Default mode -

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

FIPS mode -

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

To configure RADIUS TLS ciphers, use the following command:

```
configure radius tls cipher [[ cipher | fips_cipher | all ] on |
[ cipher | fips_cipher] off ].
```

Use the **show radius** command to show RADIUS TLS Cipher information.

Reference Identifier Support for RADIUS TLS

Version 33.4 adds support for validating RADIUS TLS server certificates using a reference identifier, enhancing compliance with Common Criteria requirements. This allows administrators to configure a reference identifier that is matched against the Subject Alternative Name (SAN) or Common Name (CN) in the server's certificate. This also ensures that the TLS session is only established if the identifier matches, improving security.

If the reference identifier configured does not match the SAN/CN of the peer certificate, the TLS session will not be established

Configuration Options:

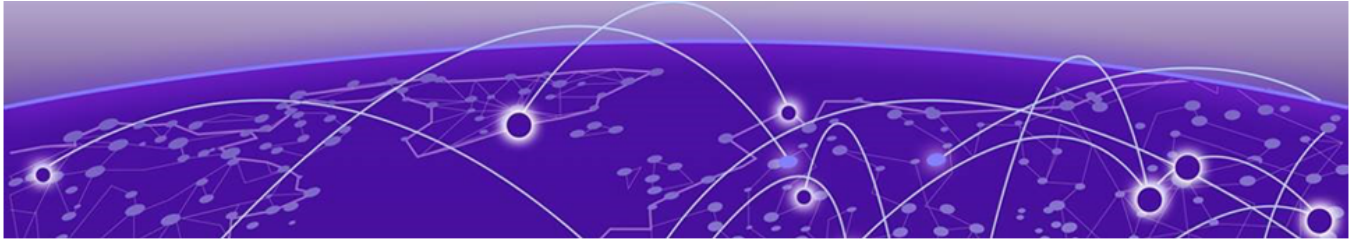
- Global reference identifier: Applies to all RADIUS TLS servers unless overridden.
- Per-server reference identifier: Takes precedence over the global setting.

Use the `configure radius` **`tls reference-identifier`** [*identifier_value* | **`none`**] command to configure a reference-identifier for RADIUS TLS connections globally. Example for PER Server Reference Identifier:

```
configure radius netlogin primary server 10.127.6.195 tls 2083 reference-identifier my-radius-server.example.com client-ip 10.127.2.33 vr VR-Mgmt shared-secret radsec
```

Supported Identifier Types:

- Only **DNS hostname** is supported.
- **Wildcard** and **IP address** identifiers are **not supported**.



REST API

REST API Overview on page 1496
Managing the OpenAPI Server on page 1503

The OpenAPI server provides RESTful API access to switch configuration and operational state, enabling integration with third-party automation and management tools such as Ansible. This chapter describes how to enable, configure, and manage the OpenAPI server on Universal Platforms.

The OpenAPI server operates alongside other web-based interfaces (Chalet, RESTCONF) and is accessible via standard HTTP/HTTPS ports using URI-based path multiplexing. When IQAgent is enabled, the OpenAPI server operates automatically to support cloud management. Users can also enable the server independently for integration with custom automation tools.

For information about using the REST API itself, including available endpoints, request formats, and response structures, refer to the OpenAPI schema documentation available at `https://<switch-ip>/rest/openapi` after enabling the server.

Note
The OpenAPI server management commands are available on Universal Platforms. Debug commands are available on all platforms. For specific information regarding platform support, see the platform-specific documentation.

REST API Overview

The OS provides a RESTful API interface that allows programmatic access to switch configuration and operational data. The REST API enables automation, integration with third-party tools, and custom application development without requiring CLI or web-based interaction.

API Architecture

The ExtremeXOS REST API is based on industry-standard HTTP/HTTPS protocols and follows RESTful design principles. API requests use standard HTTP methods (GET, POST, PUT, DELETE, PATCH) to perform operations on switch resources.

The API is accessible through the OpenAPI server, which runs alongside other web-based interfaces on the switch:

- **OpenAPI server:** Provides RESTful API access at `/rest/openapi`
- **RESTCONF:** Provides RESTCONF protocol access at `/rest/restconf`
- **Chalet:** Web-based management interface at root path `/`
- **Authentication:** Token generation endpoint at `/auth/token`

All services coexist on standard HTTP (port 80) and HTTPS (ports 443, 9443) using URI-based path multiplexing.

Accessing the API

Before using the REST API, you must enable the OpenAPI server. On Universal Platforms, use the following command:

```
enable openapi
```

Once enabled, the API is accessible at:

- `http://<switch-ip>/rest/openapi`
- `https://<switch-ip>/rest/openapi`
- `https://<switch-ip>:9443/rest/openapi`

For detailed information about managing the OpenAPI server, including enabling, disabling, restarting, and monitoring server status, see Managing the OpenAPI Server on page 1503.

Authentication

API requests require authentication using token-based authentication. The authentication workflow consists of two steps:

1. **Obtain an authentication token:** Send a POST request to `/auth/token` with your username and password:

   ```
   POST http://<switch-ip>/auth/token
   Content-Type: application/json

   {
     "username": "admin",
     "password": ""
   }
   ```

   The response includes an authentication token and its time-to-live (TTL):

   ```
   {
     "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...",
     "ttl": 86400
   }
   ```

2. **Use the token in subsequent requests:** Include the token in the `x-auth-token` header:

   ```
   GET http://<switch-ip>/rest/openapi/v0/state/system
   x-auth-token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...
   ```

Tokens remain valid for the configured TTL period (default 24 hours). After expiration, you must obtain a new token. For information about configuring token TTL, see Configurable Web Authentication Token Time-to-Live on page 1603.

Tip
Use HTTPS for all API communication to protect authentication tokens and sensitive data in transit.

API Endpoints and Operations

The REST API provides access to switch resources through structured URIs. Common resource categories include:

- **Configuration:** VLAN, port, routing, and feature configuration
- **State:** Operational status, statistics, and runtime information
- **Actions:** Operational commands like rebooting, saving configuration, or clearing counters

HTTP methods correspond to operations on resources:

- **GET:** Retrieve resource information (configuration or state)
- **POST:** Create new resources or execute actions
- **PUT:** Replace entire resource configuration
- **PATCH:** Modify specific fields within a resource
- **DELETE:** Remove resources

Example operations:

```
# Get system information
GET /rest/openapi/v0/state/system

# Get VLAN configuration
GET /rest/openapi/v0/config/vlans

# Create a VLAN
POST /rest/openapi/v0/config/vlans
Content-Type: application/json

{
  "name": "finance",
  "tag": 100
}

# Modify port configuration
PATCH /rest/openapi/v0/config/ports/1
Content-Type: application/json

{
  "admin-state": "enabled",
  "description": "Server Connection"
}
```

API Documentation and Schema

Complete API documentation, including available endpoints, request/response formats, and data models, is available through the OpenAPI schema. Once the OpenAPI server is enabled, access the schema at:

```
https://<switch-ip>/rest/openapi
```

The schema provides:

- Complete list of available API endpoints
- Request and response formats for each endpoint
- Data type definitions and validation rules
- Example requests and responses
- Error codes and troubleshooting information

The schema follows the OpenAPI Specification (formerly Swagger), which is widely supported by API development tools and client libraries.

Request and Response Format

The API uses JSON (JavaScript Object Notation) for request and response bodies. All requests and responses must include appropriate `Content-Type` headers:

Request headers:

```
Content-Type: application/json
x-auth-token: <your-token>
```

Response format:

Successful operations return HTTP status codes in the 200 range along with response data in JSON format. Common success codes include:

- **200 OK:** Successful GET, PUT, or PATCH operation
- **201 Created:** Successful POST operation that created a resource
- **204 No Content:** Successful DELETE operation or action with no return data

Error conditions return HTTP status codes in the 400 or 500 range along with error details:

- **400 Bad Request:** Invalid request format or parameters
- **401 Unauthorized:** Missing or invalid authentication token
- **404 Not Found:** Requested resource does not exist
- **500 Internal Server Error:** Server-side error processing the request

Error responses include descriptive messages to help diagnose issues:

```
{
  "error": {
    "code": 404,
    "message": "VLAN 'invalid-vlan' not found"
  }
}
```

Common Use Cases

Network Automation:

Use the REST API with automation tools like Ansible to deploy consistent configurations across multiple switches, perform automated backups, or implement configuration templates.

Custom Monitoring:

Integrate with monitoring platforms to collect switch statistics, port status, or system health metrics. Build custom dashboards or alerting systems based on real-time switch data.

Configuration Management:

Implement configuration management workflows using version control systems. Store switch configurations as JSON, track changes over time, and automate configuration deployment.

Integration with IT Systems:

Integrate switch management with ticketing systems, asset management databases, or enterprise service management platforms. Automate provisioning workflows triggered by external systems.

Custom Applications:

Develop custom applications or scripts using Python, JavaScript, or other languages to interact with switches programmatically. Build specialized tools for your specific operational needs.

Client Libraries and Tools

Several tools and libraries facilitate REST API interaction:

Command-line tools:

- **curl:** Standard command-line HTTP client for testing API endpoints
- **HTTPie:** User-friendly command-line HTTP client with JSON support

Programming language libraries:

- **Python:** requests library for HTTP operations
- **JavaScript/Node.js:** axios or fetch for API calls
- **Go:** net/http package
- **Java:** Apache HttpClient or OkHttp

API development tools:

- **Postman:** Interactive API testing and development environment
- **Swagger UI:** Interactive API documentation and testing interface
- **Insomnia:** REST API client for testing and debugging

Example using curl:

```
# Get authentication token
curl -X POST http://192.168.1.1/auth/token \
  -H "Content-Type: application/json" \
  -d '{"username":"admin","password":""}'

# Use token to get system state
curl -X GET http://192.168.1.1/rest/openapi/v0/state/system \
  -H "x-auth-token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..."
```

Example using Python requests library:

```
import requests
import json

# Get authentication token
auth_response = requests.post(
    'http://192.168.1.1/auth/token',
    json={'username': 'admin', 'password': ''}
)
token = auth_response.json()['token']

# Get system state
headers = {'x-auth-token': token}
response = requests.get(
    'http://192.168.1.1/rest/openapi/v0/state/system',
    headers=headers
)
system_info = response.json()
print(json.dumps(system_info, indent=2))
```

Best Practices

- **Use HTTPS:** Always use HTTPS in production environments to protect authentication tokens and sensitive configuration data
- **Secure token storage:** Store authentication tokens securely and never embed them in code repositories or configuration files
- **Handle token expiration:** Implement logic to detect token expiration (401 Unauthorized responses) and automatically obtain new tokens
- **Implement error handling:** Always check HTTP status codes and handle errors gracefully in your applications
- **Rate limiting awareness:** Be mindful of API request rates to avoid overwhelming the switch, especially when polling for status
- **Use appropriate HTTP methods:** Use GET for retrieving data, POST for creating resources, PATCH for updates, and DELETE for removal
- **Validate responses:** Always validate API responses before using the data in your application
- **Version awareness:** Pay attention to API version in the URI path (e.g., v0) as APIs may evolve over time
- **Test in lab:** Test API scripts and applications thoroughly in a lab environment before deploying to production
- **Monitor API server health:** Use **`show openapi`** to monitor server status and resource usage

Troubleshooting

Cannot connect to API:

- Verify the OpenAPI server is enabled: **`show openapi`**
- Check network connectivity to the switch
- Verify you're using the correct IP address and port
- Ensure web server is enabled: **`show switch management`**

Authentication failures:

- Verify username and password are correct
- Check that the token hasn't expired
- Ensure the `x-auth-token` header is included in requests
- Verify the token was copied completely without truncation

400 Bad Request errors:

- Check JSON syntax in request body
- Verify all required fields are included
- Ensure data types match schema requirements
- Check that URIs are correctly formatted

404 Not Found errors:

- Verify the resource exists on the switch
- Check URI path for typos
- Ensure the API version in the path is correct
- Verify resource names exactly match (case-sensitive)

Server errors (500 range):

- Check OpenAPI server status: **`show openapi`**
- Review system logs for errors
- Try restarting the OpenAPI server: **`restart process openapi`**
- Contact support if errors persist

Additional Resources

- OpenAPI schema documentation: *`https://<switch-ip>/rest/openapi`*
- OpenAPI server management: See Managing the OpenAPI Server on page 1503
- Token TTL configuration: See Configurable Web Authentication Token Time-to-Live on page 1603
- OpenAPI Specification: https://www.openapis.org
- REST API design principles: Industry-standard REST architectural style documentation

Managing the OpenAPI Server

Version 33.6.1 adds user-accessible commands to manage the OpenAPI server on Universal Platforms. The OpenAPI server provides RESTful API access for integration with third-party management tools such as Ansible, while maintaining compatibility with IQAgent cloud management.

OpenAPI Server Overview

The OpenAPI server provides programmatic access to switch configuration and state via a RESTful API. Previously managed exclusively by IQAgent for internal cloud applications, the OpenAPI server is now available for customer use.

The OpenAPI server operates alongside other web-based interfaces on the switch:

- **Chalet:** Web-based management interface (root path `/`)
- **OpenAPI:** RESTful API interface (`/rest/openapi`)
- **RESTCONF:** RESTCONF protocol interface (`/rest/restconf`)
- **Authentication:** Token generation (`/auth/token`)

The web server multiplexes these applications based on URI paths, allowing all services to coexist on standard HTTP (port 80) and HTTPS (ports 443, 9443). There are no changes to existing **`enable web http`** or **`enable web https`** commands.

Enabling the OpenAPI Server

To enable the OpenAPI server for third-party tool integration:

```
enable openapi
```

Once enabled, the OpenAPI server accepts RESTful API requests at:

- `http://<switch-ip>/rest/openapi`
- `https://<switch-ip>/rest/openapi`
- `https://<switch-ip>:9443/rest/openapi`

The OpenAPI server state persists across reboots.

Disabling the OpenAPI Server

To disable the OpenAPI server when it is not needed:

```
disable openapi
```

This stops the OpenAPI server process and prevents API access via the `/rest/openapi` URI path.

IQAgent Override:

When IQAgent is enabled, the OpenAPI server is required for cloud connectivity and cannot be disabled by the user. Attempting to disable the server when IQAgent is enabled displays:

```
Note: The OpenAPI server is required by IQAgent and will remain operational
while IQAgent is enabled.
```

The server remains running and the state shows as "Enabled (by IQAgent)". To disable the OpenAPI server when IQAgent is enabled, you must first disable IQAgent.

Restarting the OpenAPI Server

To restart the OpenAPI server process:

```
restart process openapi
```

Restarting may be necessary to:

- Apply configuration changes that require a restart
- Recover from an unhealthy or failed server state
- Clear temporary issues or reset connections
- Apply debug mode changes

During the restart, API access via the `/rest/openapi` URI path is temporarily unavailable. Schedule restarts during maintenance windows when possible to minimize impact on integrated management tools.

Viewing OpenAPI Server Status

To view the current state and operational status of the OpenAPI server:

```
show openapi
```

Sample output when enabled by user:

```
State            : Enabled (by user)
Server Status    : Healthy
Version          : 1.4.12.16
Schema Version   : 0.2.0
```

Sample output when IQAgent requires OpenAPI:

```
State            : Enabled (by IQAgent)
Server Status    : Healthy
Version          : 1.4.12.16
Schema Version   : 0.2.0
```

Sample output when disabled:

```
State            : Disabled
Server Status    : Stopped
Version          : 1.4.12.16
Schema Version   : 0.2.0
```

Output Fields:

- **State:** Whether the server is enabled (and by whom) or disabled
  - Enabled (by user) - Enabled via user command
  - Enabled (by IQAgent) - Required by IQAgent
  - Enabled (by IQAgent and user) - Both enabled it
  - Disabled - Server is not running
- **Server Status:** Current operational status
  - Healthy - Running normally
  - Unhealthy - Running but experiencing issues
  - Booting - Starting up
  - Stopped - Not running
  - Failed - Failed to start or crashed
- **Version:** Semantic version of the OpenAPI application (N/A if unavailable)
- **Schema Version:** Semantic version of the NOS-OpenAPI YAML schema (N/A if unavailable)

To display HTTPS (SSL) related port information, use the `show` **`web`** command.

```
# show web
HTTP
    Access         : Disabled
    Port number(s) : 80
    Access profile : not set

HTTPS
    Access         : Enabled
    Port number(s) : 443, 9443

Authentication token default TTL: 60 seconds
```

Integration with Management Tools

The OpenAPI server enables integration with third-party automation and management tools. Common use cases include:

- **Ansible:** Use Ansible playbooks to automate switch configuration and management via the OpenAPI
- **Python scripts:** Develop custom automation scripts using Python with RESTful API calls
- **Monitoring systems:** Integrate with monitoring platforms to collect switch state and performance data
- **Configuration management:** Implement configuration management workflows using standard REST APIs

Refer to the OpenAPI schema documentation for details on available API endpoints, request formats, and response structures.

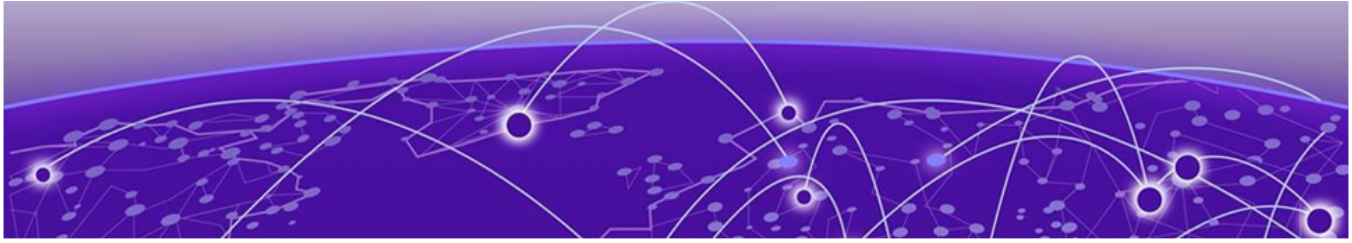
Authentication

API requests to the OpenAPI server require authentication. Use token-based authentication by first obtaining an authentication token from the `/auth/token` endpoint, then include the token in subsequent API requests via the `x-auth-token` header.

See Configurable Web Authentication Token Time-to-Live on page 1603 for information on configuring token time-to-live settings.

Best Practices

- Enable the OpenAPI server only when actively using third-party tools that require API access
- Use HTTPS (ports 443 or 9443) for API access to ensure encrypted communication
- Implement proper authentication and authorization in your API client applications
- Monitor server status when integrating new tools to detect potential issues early
- Schedule server restarts during maintenance windows to minimize impact on automation workflows
- Keep debug mode disabled during normal operation to avoid excessive logging



RIP

IGPs Overview on page 1507
Overview of RIP on page 1508
Route Redistribution on page 1510
RIP Configuration Example on page 1512

This chapter assumes that you are already familiar with IP unicast routing. If not, refer to the following publications for additional information:

- RFC 1058—Routing Information Protocol (RIP)
- RFC 1723—RIP Version 2
- *Interconnections: Bridges and Routers* by Radia Perlman. ISBN 0-201-56332-0. Published by Addison-Wesley Publishing Company.

Note
RIP is available on platforms with an Edge (or higher) or Base (or higher) license. For specific information regarding RIP licensing, see the licensing guide.

IGPs Overview

The switch supports the use of the following interior gateway protocols (IGPs):

- *RIP (Routing Information Protocol)*
- *OSPF (Open Shortest Path First)*
- Intermediate System-Intermediate System (IS-IS)

RIP is a distance-vector protocol, based on the Bellman-Ford (or distance-vector) algorithm. The distance-vector algorithm has been in use for many years and is widely deployed and understood.

OSPF and IS-IS are link-state protocols, based on the Dijkstra link-state algorithm. OSPF and IS-IS are newer IGPs and solve a number of problems associated with using RIP on today's complex networks.

Note
RIP can be enabled on a *VLAN (Virtual LAN)* with either OSPF or IS-IS. OSPF and IS-IS cannot be enabled on the same VLAN.

RIP is described in this chapter, OSPF is described in OSPF and IS-IS is described in IS-IS.

RIP Versus OSPF and IS-IS

The distinction between *RIP* and the *OSPF* and IS-IS link-state protocols lies in the fundamental differences between distance-vector protocols and link-state protocols.

Using a distance-vector protocol, each router creates a unique routing table from summarized information obtained from neighboring routers. Using a link-state protocol, every router maintains an identical routing table created from information obtained from all routers in the autonomous system (AS). Each router builds a shortest path tree, using itself as the root. The link-state protocol ensures that updates sent to neighboring routers are acknowledged by the neighbors, verifying that all routers have a consistent network map.

Advantages of RIP, OSPF, and IS-IS

The biggest advantage of using *RIP* is that it is relatively simple to understand and to implement, and it has been the de facto routing standard for many years.

RIP has a number of limitations that can cause problems in large networks, including the following:

- A limit of 15 hops between the source and destination networks
- A large amount of bandwidth taken up by periodic broadcasts of the entire routing table
- Slow convergence
- Routing decisions based on hop count; no concept of link costs or delay
- Flat networks; no concept of areas or boundaries

OSPF and IS-IS offer many advantages over RIP, including the following:

- No limitation on hop count
- Route updates multicast only when changes occur
- Faster convergence
- Support for load balancing to multiple routers based on the actual cost of the link
- Support for hierarchical topologies where the network is divided into areas

The details of RIP are explained later in this chapter.

Overview of RIP

RIP is an IGP first used in computer routing in the Advanced Research Projects Agency Network (ARPAnet) as early as 1969. It is primarily intended for use in homogeneous networks of moderate size.

To determine the best path to a distant network, a router using RIP always selects the path that has the least number of hops. Each router that data must traverse is considered to be one hop.

Routing Table

The routing table in a router using *RIPng (Routing Information Protocol Next Generation)* contains an entry for every known destination network.

Each routing table entry contains the following information:

- IP address and prefix length of the destination network
- Metric (hop count) to the destination network
- IP address of the next hop router, if the destination is not directly connected
- Interface for the next hop
- Timer that tracks the amount of time since the entry was last updated
- A flag that indicates if the entry is a new one since the last update
- The source of the route, for example, static, RIPng, *OSPFv3 (Open Shortest Path First version 3)*, etc.

The router exchanges an update message with each neighbor every 30 seconds (default value), or when there is a change to the overall routed topology (also called *triggered updates*). If a router does not receive an update message from its neighbor within the route timeout period (180 seconds by default), the router assumes the connection between it and its neighbor is no longer available.

Split Horizon

Split horizon is a scheme for avoiding problems caused by including routes in updates sent to the router from which the route was learned.

Split horizon omits routes learned from a neighbor in updates sent to that neighbor.

Poison Reverse

Like split horizon, poison reverse is a scheme for eliminating the possibility of loops in the routed topology.

In this case, a router advertises a route over the same interface that supplied the route, but the route uses a hop count of 16, which defines that router as unreachable.

Triggered Updates

Triggered updates occur whenever a router changes the metric for a route.

The router is required to send an update message immediately, even if it is not yet time for a regular update message to be sent. This generally results in faster convergence, but may also result in more *RIPng*-related traffic.

Route Advertisement of VLANs

Virtual LANs (VLANs) that are configured with an IP address but are configured to not route IP or are not configured to run *RIP*, do not have their subnets advertised by RIP.

RIP advertises only those VLANs that are configured with an IP address, are configured to route IP, and run RIP.

RIP Version 1 Versus RIP Version 2

A new version of *RIP*, called RIP version 2, expands the functionality of RIP version 1 to include the following:

- Variable-length subnet masks (VLSMs).
- Support for next-hop addresses, which allows for optimization of routes in certain environments.
- Multicasting.

  RIP version 2 packets can be multicast instead of being broadcast, reducing the load on hosts that do not support routing protocols.

Note
If you are using RIP with supernetting/Classless Inter-Domain Routing (CIDR), you must use RIPv2 only.

Route Redistribution

More than one routing protocol can be enabled simultaneously on the switch.

Route redistribution allows the switch to exchange routes, including static routes, between the routing protocols. Figure 185 is an example of route redistribution between an OSPF AS and a RIP AS.

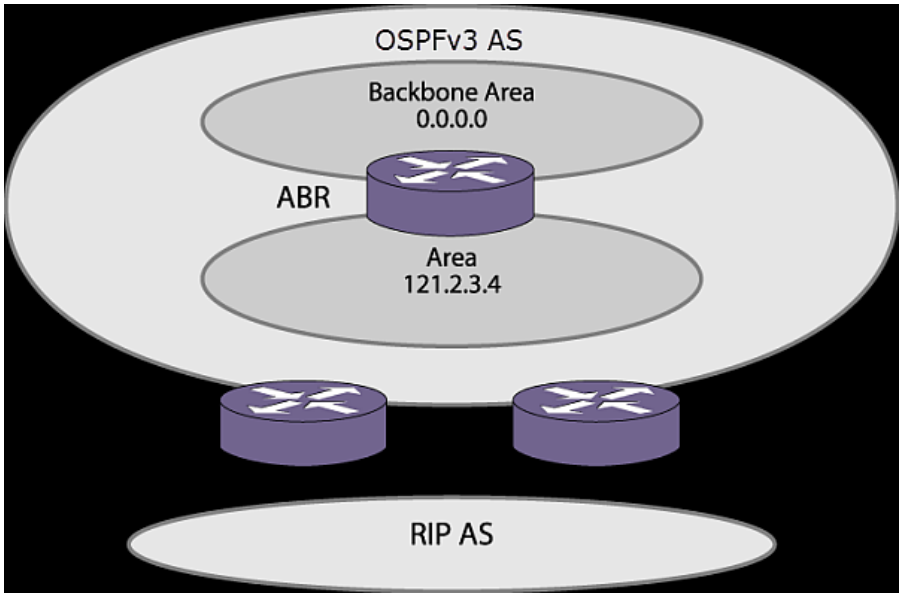


Figure 185: OSPF Route Redistribution

Configuring Route Redistribution

Exporting routes from one protocol to another and in the reverse direction are discrete configuration functions. For example, to run IS-IS and *RIP* simultaneously, you must first configure both protocols and then verify the independent operation of each. Then you can configure the routes to export from IS-IS to RIP and the routes to export from RIP to IS-IS. Likewise, for any other combinations of protocols, you must separately configure each to export routes to the other.

This section describes how to inject routing information learned from other IP routing protocols into an IS-IS domain, thereby advertising their reachability in the IS-IS domain. These routes are included in the locally originated LSPs. For information on exporting routes learned in IS-IS to another routing protocol, see the chapter that describes the destination routing protocol.

When you export routes from another protocol to IS-IS, the metric and the type of metric are configured. You can also configure a policy to filter out unwanted external routes.

- To enable or disable the exporting of *BGP (Border Gateway Protocol)*, *OSPF*, RIP, static, and direct (interface) IPv4 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export {ipv4} route-type [policy | metric mvalue {metric-type [internal | external]}] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export {ipv4} route-type
  ```

- To enable or disable the exporting of *OSPFv3*, *RIPng*, static, and direct (interface) IPv6 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export ipv6 route-type [policy | metric mvalue] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export ipv6 route-type
  ```

  The cost metric is inserted for all *OSPF*, RIP, static, and direct routes injected into IS-IS.

  The same cost and type values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route. The exported routes can also be filtered using policies.

- Verify the configuration using the command:

  ```
  show isis area [area_name | all]
  ```

Redistribute Routes into RIP

- Enable or disable the exporting of static, direct, *BGP*-learned, and *OSPF*-learned routes into the *RIP* domain.

  ```
  enable rip export [bgp | direct | e-bgp | i-bgp | ospf | ospf-extern1 | ospf-extern2 | ospf-inter | ospf-intra | static | isis | isis-level-1| isis-level-1-external | isis-level-2 | isis-level-2-external ] [cost number {tag number} | policy policy-name]
  ```

  ```
  disable rip export [bgp | direct | e-bgp | i-bgp | ospf | ospf-extern1 | ospf-extern2 | ospf-inter | ospf-intra | static | isis
  ```

| isis-level-1| isis-level-1-external | isis-level-2 | isis-level-2-external]

These commands enable or disable the exporting of static, direct, and OSPF-learned routes into the RIP domain.

- You can choose which types of OSPF routes are injected, or you can simply choose ospf, which will inject all learned OSPF routes regardless of type.

  The default setting is disabled.

RIP Configuration Example

Figure 186 on page 1512 illustrates a switch with three VLANs defined as follows:

- Finance
  - Protocol-sensitive *VLAN* using the IP protocol.
  - All ports on slots 1 and 3 have been assigned.
  - IP address 192.207.35.1.
- Personnel
  - Protocol-sensitive VLAN using the IP protocol.
  - All ports on slots 2 and 4 have been assigned.
  - IP address 192.207.36.1.
- MyCompany
  - Port-based VLAN.
  - All ports on slots 1 through 4 have been assigned.

This example does use protocol-sensitive VLANs that admit only IP packets. This is not a common requirement for most networks. In most cases, VLANs will admit different types of packets to be forwarded to the hosts and servers on the network.

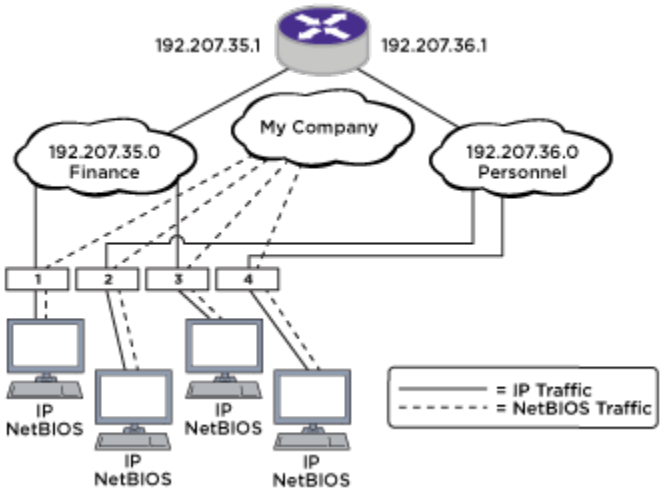


Figure 186: RIP Configuration Example

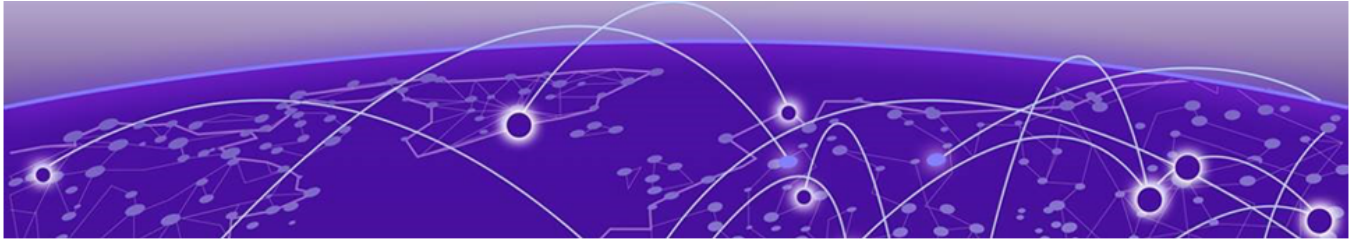
The stations connected to the system generate a combination of IP traffic and NetBIOS traffic. The IP traffic is filtered by the protocol-sensitive VLANs. All other traffic is directed to the VLAN MyCompany.

In this configuration, all IP traffic from stations connected to slots 1 and 3 have access to the router by way of the VLAN Finance. Ports on slots 2 and 4 reach the router by way of the VLAN Personnel. All other traffic (NetBIOS) is part of the VLAN MyCompany.

The example in Figure 186 is configured as follows:

```
create vlan Finance
create vlan Personnel
create vlan MyCompany
configure Finance protocol ip
configure Personnel protocol ip
configure Finance add port 1:*,3:*
configure Personnel add port 2:*,4:*
configure MyCompany add port all
configure Finance ipaddress 192.207.35.1
configure Personnel ipaddress 192.207.36.1
enable ipforwarding
configure rip add vlan all
enable rip
```

More commonly, NetBIOS traffic would be allowed on the Finance and Personnel VLANs, but this example shows how to exclude that traffic. To allow the NetBIOS traffic (or other type of traffic) along with the IP traffic, remove the `configure finance protocol ip` and `configure Personnel protocol ip` commands from the example.



RIPng

RIPng Overview on page 1514
RIPng Routing on page 1515
Route Redistribution on page 1517
RIPng Configuration Example on page 1518

This chapter describes the *RIPng (Routing Information Protocol Next Generation)* interior gateway protocol for IPv6 networks. It provides the various protocol schemes available, commands for configuring redistribution, and an example for configuring multiple protocols on one switch. This chapter assumes that you are already familiar with IP unicast routing. If not, refer to the following publication for additional information:

- RFC 2080—RIPng for IPv6

Note
RIPng is available on platforms with an Edge (or higher) or Base (or higher) license. For specific information regarding RIPng licensing. see the Switch Engine v33.6.1 Licensing Guide document.

RIPng Overview

RIPng is an interior gateway protocol (IGP) developed for IPv6 networks.

The analogous protocol used in IPv4 networks is called *RIP (Routing Information Protocol)*. Like RIP, RIPng is a relatively simple protocol for the communication of routing information among routers. Many concepts and features of RIPng are directly parallel to those same features in RIP.

RIPng is a distance-vector protocol, based on the Bellman-Ford (or distance-vector) algorithm. The distance-vector algorithm has been in use for many years and is widely deployed and understood. The other common IGPs for IPv6 are *OSPFv3 (Open Shortest Path First version 3)* and Intermediate System-Intermediate System (IS-IS), which are link-state protocols.

Note
RIPng can be enabled on a *VLAN (Virtual LAN)* with either OSPFv3 or IS-IS. OSPFv3 and ISIS cannot be enabled on the same VLAN.

RIPng versus OSPFv3 and IS-IS

The distinction between *RIPng* and the link-state protocols (*OSPFv3* and IS-IS) lies in the fundamental differences between distance-vector protocols (RIPng) and link-state protocols.

Using a distance-vector protocol, each router creates a unique routing table from summarized information obtained from neighboring routers. Using a link-state protocol, every router maintains an identical routing table created from information obtained from all routers in the autonomous system. Each router builds a shortest path tree, using itself as the root. The link-state protocol ensures that updates sent to neighboring routers are acknowledged by the neighbors, verifying that all routers have a consistent network map.

Advantages of RIPng, OSPFv3, and IS-IS

The biggest advantage of using *RIPng* is that it is relatively simple to understand and implement, and it has been the de facto routing standard for many years.

RIPng has a number of limitations that can cause problems in large networks, including the following:

- A limit of 15 hops between the source and destination networks
- A large amount of bandwidth taken up by periodic broadcasts of the entire routing table
- Slow convergence
- Routing decisions based on hop count; no concept of link costs or delay
- Flat networks; no concept of areas or boundaries

OSPFv3 and IS-IS offer many advantages over RIPng, including the following:

- No limitation on hop count
- Route updates multicast only when changes occur
- Faster convergence
- Support for load balancing to multiple routers based on the actual cost of the link
- Support for hierarchical topologies where the network is divided into areas

The details of RIPng are explained later in this chapter.

RIPng Routing

RIPng is primarily intended for use in homogeneous networks of moderate size.

To determine the best path to a distant network, a router using RIPng always selects the path that has the least number of hops. Each router that data must traverse is considered to be one hop.

Routing Table

The routing table in a router using *RIPng* contains an entry for every known destination network.

Each routing table entry contains the following information:

- IP address and prefix length of the destination network
- Metric (hop count) to the destination network
- IP address of the next hop router, if the destination is not directly connected
- Interface for the next hop
- Timer that tracks the amount of time since the entry was last updated
- A flag that indicates if the entry is a new one since the last update
- The source of the route, for example, static, RIPng, *OSPFv3*, etc.

The router exchanges an update message with each neighbor every 30 seconds (default value), or when there is a change to the overall routed topology (also called *triggered updates*). If a router does not receive an update message from its neighbor within the route timeout period (180 seconds by default), the router assumes the connection between it and its neighbor is no longer available.

Split Horizon

Split horizon is a scheme for avoiding problems caused by including routes in updates sent to the router from which the route was learned.

Split horizon omits routes learned from a neighbor in updates sent to that neighbor.

Poison Reverse

Like split horizon, poison reverse is a scheme for eliminating the possibility of loops in the routed topology.

In this case, a router advertises a route over the same interface that supplied the route, but the route uses a hop count of 16, which defines that router as unreachable.

Triggered Updates

Triggered updates occur whenever a router changes the metric for a route.

The router is required to send an update message immediately, even if it is not yet time for a regular update message to be sent. This generally results in faster convergence, but may also result in more *RIPng*-related traffic.

Route Advertisement of VLANs

Virtual LANs (VLANs) that are configured with an IP address but are configured to not route IP or are not configured to run *RIP*, do not have their subnets advertised by RIP.

RIP advertises only those VLANs that are configured with an IP address, are configured to route IP, and run RIP.

Route Redistribution

More than one routing protocol can be enabled simultaneously on the switch.

Route redistribution allows the switch to exchange routes, including static routes, between the routing protocols. Figure 187 is an example of route redistribution between an OSPF AS and a RIP AS.

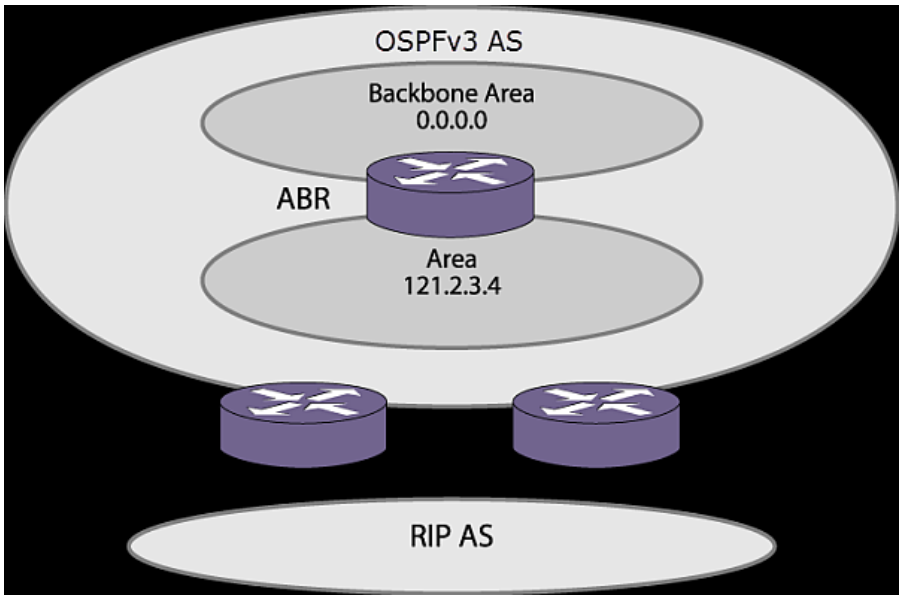


Figure 187: OSPF Route Redistribution

Configuring Route Redistribution

Exporting routes from one protocol to another and in the reverse direction are discrete configuration functions. For example, to run IS-IS and *RIP* simultaneously, you must first configure both protocols and then verify the independent operation of each. Then you can configure the routes to export from IS-IS to RIP and the routes to export from RIP to IS-IS. Likewise, for any other combinations of protocols, you must separately configure each to export routes to the other.

This section describes how to inject routing information learned from other IP routing protocols into an IS-IS domain, thereby advertising their reachability in the IS-IS domain. These routes are included in the locally originated LSPs. For information on exporting routes learned in IS-IS to another routing protocol, see the chapter that describes the destination routing protocol.

When you export routes from another protocol to IS-IS, the metric and the type of metric are configured. You can also configure a policy to filter out unwanted external routes.

- To enable or disable the exporting of *BGP (Border Gateway Protocol)*, *OSPF (Open Shortest Path First)*, RIP, static, and direct (interface) IPv4 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export {ipv4} route-type [policy | metric mvalue {metric-type [internal | external]}] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export {ipv4} route-type
  ```

- To enable or disable the exporting of *OSPFv3*, *RIPng*, static, and direct (interface) IPv6 routes to IS-IS, use the following commands:

  ```
  enable isis area area_name export ipv6 route-type [policy | metric mvalue] {level[1| 2 | both-1-and-2]}
  ```

  ```
  disable isis area area_name export ipv6 route-type
  ```

  The cost metric is inserted for all *OSPF*, RIP, static, and direct routes injected into IS-IS.

  The same cost and type values can be inserted for all the export routes, or policies can be used for selective insertion. When a policy is associated with the export command, the policy is applied on every exported route. The exported routes can also be filtered using policies.

- Verify the configuration using the command:

  ```
  show isis area [area_name | all]
  ```

Redistributing Routes into RIPng

- Eable or disable the exporting of static, direct, or other protocol-learned routes into the *RIPng* domain using the following commands:

  ```
  enable ripng export [direct | ospfv3 | ospfv3-extern1 | ospfv3-extern2 | ospfv3-inter | ospfv3-intra | static | isis | isis-level-1| isis-level-1-external | isis-level-2| isis-level-2-external | bgp | e-bgp i-bgp] [cost number {tag number} | policy policy-name]
  ```

  ```
  disable ripng originate-default
  ```

  ```
  enable ripng originate-defalut {always} {cost value}
  ```

  These commands enable or disable the exporting of static, direct, and OSPFv3-learned routes into the RIPng domain.

- You can choose which types of OSPFv3 routes are injected, or you can simply choose ospf, which will inject all learned OSPFv3 routes regardless of type. The default setting is disabled.

RIPng Configuration Example

The following configuration is similar to the example in the *RIP* chapter, but uses IPv6 addresses. It illustrates a switch with three VLANs defined as follows:

- Finance
  - All ports on slots 1 and 3 have been assigned.

    - IP address 2001:db8:35::1/48.
- Personnel
    - All ports on slots 2 and 4 have been assigned.
    - IP address 2001:db8:36::1/48.
- MyCompany
    - Port-based *VLAN*.
    - All ports on slots 1 through 4 have been assigned.

The stations connected to the system generate a combination of IPv6 traffic and NetBIOS traffic.

In this configuration, all traffic from stations connected to slots 1 and 3 have access to the router by way of the VLAN Finance. Ports on slots 2 and 4 reach the router by way of the VLAN Personnel. All traffic (NetBIOS and IPv6) is part of the VLAN MyCompany.

The example is configured as follows:

```
create vlan Finance
create vlan Personnel
create vlan MyCompany
configure Finance add port 1:*,3:*
configure Personnel add port 2:*,4:*
configure MyCompany add port all
configure Finance ipaddress 2001:db8:35::1/48
configure Personnel ipaddress 2001:db8:36::1/48
enable ipforwarding ipv6
configure ripng add vlan Finance
configure ripng add vlan Personnel
enable ripng
```

Routing Policies

Routing Policies Overview on page 1520
Routing Policy File Syntax on page 1521
Applying Routing Policies on page 1527
Policy Examples on page 1527

This chapter provides information about Routing Policies. It includes an overview, specific information about Routing Policy File Syntax, how to apply Routing Policies, and offers some Policy examples.

Routing Policies Overview

Routing policies are used to control the advertisement or recognition of routes communicated by routing protocols, such as *RIP (Routing Information Protocol)*, *OSPF (Open Shortest Path First)*, Intermediate System-Intermediate System (IS-IS) and *BGP (Border Gateway Protocol)*.

Routing policies can be used to “hide” entire networks or to trust only specific sources for routes or ranges of routes. The capabilities of routing policies are specific to the type of routing protocol involved, but these policies are sometimes more efficient and easier to implement than access lists.

Routing policies can also modify and filter routing information received and advertised by a switch.

A similar type of policy is an *ACL (Access Control List)* policy, used to control, at the hardware level, the packets accessing the switch. ACL policy files and routing policy files are both handled by the policy manager and the syntax for both types of files is checked by the policy manager.

Note
Although ExtremeXOS does not prohibit mixing ACL and routing type entries in a policy file, it is strongly recommended that you do not mix the entries, and you use separate policy files for ACL and routing policies.

Routing Policy File Syntax

A routing policy file contains one or more policy rule entries. Each routing policy entry consists of:

- A policy entry rule name, unique within the same policy.
- Zero or one match type. If no type is specified, the match type is all, so all match conditions must be satisfied.
- Zero or more match conditions. If no match condition is specified, then every routing entity matches.
- Zero or more actions. If no action is specified, the packet is permitted by default.

Each policy entry in the file uses the following syntax:

```
entry <routingrulename>{
    if <match-type> {
        <match-conditions>;
    } then {
        <action>;
    }
}
```

The following is an example of a policy entry:

```
entry ip_entry {
    if match any {
        nlri  10.203.134.0/24;
        nlri  10.204.134.0/24;
    } then {
        next-hop   192.168.174.92;
        origin   egp;
    }
}
```

Policy entries are evaluated in order, from the beginning of the file to the end, as follows:

- If a match occurs, the action in the then statement is taken:
  - if the action contains an explicit permit or deny, the evaluation process terminates.
  - if the action does not contain an explicit permit or deny, the action is an implicit permit, and the evaluation process terminates.
- If a match does not occur, the next policy entry is evaluated.
- If no match has occurred after evaluating all policy entries, the default action is deny.

Often a policy has a rule entry at the end of the policy with no match conditions. This entry matches anything not otherwise processed, so that the user can specify an action to override the default deny action.

Policy match type, match conditions and action statements are discussed in the following sections:

- Policy Match Type on page 1522
- Policy Match Conditions on page 1522

- Autonomous System Expressions on page 1523
- Policy Action Statements on page 1525

Policy Match Type

The two possible choices for the match type are:

- match all—All the match conditions must be true for a match to occur. This is the default.
- match any—If any match condition is true, then a match occurs.

Policy Match Conditions

Table 134 lists the possible policy entry match conditions.

Table 134: Policy Match Conditions

Match Condition	Description
as-path [<as-number> | <as-path-regular-expression>];	Where <as-number> is a valid 2-byte AS number in the range of 1 to 65535 or a 4-byte AS number in the range of 65536 to 4294967294. Where <as-path-regular-expression> is a multi-character regular expression (with 2-byte unsigned Integer being an Atom). Regular expression will consist of the AS-Numbers and various regular expression symbols. Regular expressions must be enclosed in double quotes ("").
community [no-advertise | no-export | no-export-subconfed | number <community_num> | <community_regular_expression> | <as_num> : <num>];	Where no-advertise, no-export and no-export-subconfed are the standard communities defined by RFC. <community_num> is a four-byte unsigned integer, <as_num> is a two-byte or four-byte AS-Number and <num> is the 2-bytes community number. Community regular expression is a multi-character regular expression (with four byte unsigned integer being an Atom). Regular expression is enclosed in double quotes ("").
med <number>;	Where <number> is a 4-byte unsigned integer.
next-hop [<ipaddress> | <ipaddress-regular-expression>];	Where <ipaddress> is a valid IP address in dotted decimal format.

Table 134: Policy Match Conditions (continued)

Match Condition	Description
nlri [*ipaddress* | any] *mask-length* {exact}; nlri [*ipaddress* | any] mask *mask* {exact}; nlri [*ipv6address* | any-ipv6]/ *mask-length* {exact};	Where *ipaddress* and *mask* are IPv4 addresses and masks, *mask-length* is an integer with maximum value of 32 for IPv4 addresses. The keyword any matches any IPv4 address with a given (or larger) mask/ mask-length. Similarly *ipv6address* is an IPv6 address and *masklength* is an integer with a maximum value of 128 for IPv6 addresses. The keyword any-ipv6 matches any IPv6 address with a given (or larger) mask-length.
origin [igp | egp | incomplete];	Where igp, egp and incomplete are the *BGP* route origin values.
tag <number>;	Where <number> is a 4-byte unsigned number. **Note:** The tag match condition only works with export policy.
route-origin [direct | static | icmp | egp | ggp | hello | rip | isis | esis | cisco-igrp | ospf | bgp | idrp | dvmrp | mospf | pim-dm | pim-sm | ospf-intra | ospf-inter | ospf-extern1 | ospf-extern2 | bootp | e-bgp | i-bgp | mbgp | i-mbgp | e-mbgp | isis-level-1 | isis-level-2 | isis-level-1-external | isis-level-2-external];	Matches the origin (different from BGP route origin) of a route. A match statement "route-origin bgp" will match routes whose origin are "I-bgp" or "e-bgp" or "I-mbgp" or "e-mbgp". Similarly, the match statement "route-origin ospf" will match routes whose origin is "ospf-inta" or "ospf-inter" or "ospf-as-external" or "ospf-extern-1" or "ospf-extern-2"

Note
When entering an AS number in a policy file, you must enter a unique 2-byte or 4-byte AS number. The transition AS number, AS 23456, is not supported in policy files.

Autonomous System Expressions

The AS-path keyword uses a regular expression string to match against the autonomous system (AS) path. Table 135 lists the regular expressions that can be used in the match conditions for BGP AS path and community. It also shows examples of regular expressions and the AS paths they match.

Table 135: AS Regular Expression Notation

Character	Definition
N	As number
N1 - N2	Range of AS numbers, where N1 and N2 are AS numbers and N1 < N2

Table 135: AS Regular Expression Notation (continued)

Character	Definition
[Nx ... Ny]	Group of AS numbers, where Nx and Ny are AS numbers or a range of AS numbers
[^Nx ... Ny]	Any AS numbers other than the ones in the group
.	Matches any number
^	Matches the beginning of the AS path
$	Matches the end of the AS path
–	Matches the beginning or end, or a space
-	Separates the beginning and end of a range of numbers
*	Matches 0 or more instances
+	Matches 1 or more instances
?	Matches 0 or 1 instance
{	Start of AS SET segment in the AS path
}	End of AS SET segment in the AS path
(	Start of a confederation segment in the AS path
)	End of a confederation segment in the AS path

Table 136: Policy Regular Expression Examples

Attribute	Regular Expression	Example Matches
AS path is 64496	“64496”	64496
Zero or more occurrences of AS number 1234	“64496*”	64496 64496 64496
Start of AS path set	“64496 64500 { 64505”	64496 64500 64505 { 64511 64509 64496 64500 { 64505 64507
End of AS path set	“64500 } 64505”	64500 } 64505 56
Path that starts with 99 followed by 34	“^64511 64505 “	64511 64505 45
Path that ends with 99	“64511 $”	45 66 64511
Path of any length that begins with AS numbers 64496 64497 64498	“64496 64497 64498 .*”	64496 64497 64498 64499 64500 64501 64502 64503 64504
Path of any length that ends with AS numbers 64496 64497 64498	“.* 64496 64497 64498 $”	64496 64497 64498 64502 64503 64504 64496 64497 64498

Following are additional examples of using regular expressions in the AS-Path statement.

The following AS-Path statement matches AS paths that contain only (begin and end with) AS number 64511:

```
as-path "^64511$"
```

The following AS-Path statement matches AS paths beginning with AS number 64500, ending with AS number 64511, and containing no other AS paths:

```
as-path "^64500 64511$"
```

The following AS-Path statement matches AS paths beginning with AS number 64496, followed by any AS number from 65500 - 65505, and ending with either AS number 65507, 65509, or 65511:

```
as-path "^64496 65500-65505 [65507 65509 65511]$"
```

The following AS-Path statement matches AS paths beginning with AS number 65511 and ending with any AS number from 65500 - 65505:

```
as-path "65511 [65500-65505]$"
```

The following AS-Path statement matches AS paths beginning with AS number 65511 and ending with any additional AS number, or beginning and ending with AS number 65511:

```
as-path "65511 .?"
```

Policy Action Statements

Table 137 lists policy action statements. These are the actions taken when the policy match conditions are met in a policy entry.

Table 137: Policy Actions

Action	Description
as-path "<as_num> {<as_num1> <as_num2> <as_num3> .... <as_numN>}";	Prepends the entire list of as-numbers to the as-path of the route.
community set [no-advertise | no-export | noexport- subconfed | <community_num> | <as_num> : <community_num>];	Replaces the existing community attribute of a route by the community specified by the action statement. Community must be enclosed in double quotes ("").
community [add | delete] [no-advertise | no-export | no-export-subconfed | <community_num> {<community_num1> <community_num2> .... <community_numN>} | <as_num> : <community_num> {<as_num1> <community_num1> <as_num2> <community_num2> ....}];	Adds/deletes communities to/from a route's community attribute. Communities must be enclosed in double quotes ("").
community remove;	Strips off the entire community attribute from a route. Communities must be enclosed in double quotes ("").
cost <cost(0-4261412864)>;	Sets the cost/metric for a route.

Table 137: Policy Actions (continued)

Action	Description
cost-type {ase-type-1 | ase-type-2 | external | internal};	Sets the cost type for a route.
dampening half-life <minutes (1-45)> reuse-limit <number (1-20000)> suppress-limit <number (1-20000)> max-suppress <minutes (1-255)>;	Sets the *BGP* route flap dampening parameters.
deny;	Denies the route.
local-preference <number>;	Sets the BGP local preference for a route.
med {add | delete} <number>;	Performs MED arithmetic. Add means the value of the MED in the route will be incremented by <number>, and delete means the value of the MED in the route will be decremented by <number>.
med {internal | remove};	Internal means that the Interior Gateway Protocol (IGP) distance to the next hop will be taken as the MED for a route. Remove means take out the MED attribute from the route.
med set <number>;	Sets the MED attribute for a route.
next-hop <ipv4 address | ipv6 address>;	Sets the next hop attribute for a route.
nlri [<ipaddress> | any]/<mask-length> {exact};nlri [<ipaddress> | any] mask <mask> {exact};	These set statements are used for building a list of IP addresses. This is used by PIM to set up the RP list.
origin {igp | egp | incomplete};	Sets the BGP route origin values.
permit;	Permits the route.
tag <number>;	Sets the tag number for a route.
weight <number>	Sets the weight for a BGP route.

Note

Multiple communities couldn't be used in "community set" attribute in the BGP policy file. The way to set multiple communities in BGP policy file could be accomplished by two set of attributes as given in below example:

```
entry permit-anything-else {
   if {
   } then {
     community set "2342:6788";
     community add "2342:6789 2342:6790";
 permit;
   }
}
```

Applying Routing Policies

To apply a routing policy, use the command appropriate to the client. Different protocols support different ways to apply policies, but there are some generalities.

Commands that use the keyword import-policy are used to change the attributes of routes installed into the switch routing table by the protocol. These commands cannot be used to determine the routes to be added to the routing table. The following are examples for the *BGP* and *RIP* protocols:

```
configure bgp import-policy [policy-name | none]

configure rip import-policy [policy-name | none]
```

Commands that use the keyword route-policy control the routes advertised or received by the protocol. For BGP and RIP, here are some examples:

```
configure bgp neighbor [remoteaddr | all] {address-family [ipv4-unicast
| ipv4-multicast |ipv6-unicast | ipv6-multicast |vpnv4]} route-policy
[in | out] [none | policy]

configure bgp peer-group peer-group-name {address-family [ipv4-unicast |
ipv4-multicast |ipv6-unicast | ipv6-multicast |vpnv4]} route-policy [in
|out] [none | policy]

configure rip vlan [vlan_name | all] route-policy [in | out] [policy-
name | none]
```

Other examples of commands that use routing policies include:

```
configure ospf area area-identifier external-filter [policy-map |none]

configure ospf add vlan [vlan-name | all] area area-identifier {passive}
{vr vrf_name}

configure rip vlan [vlan_name | all] trusted-gateway [policy-name |
none]
```

To remove a routing policy, use the **none** option in the command.

Note
When exporting a route into BGP with an export policy, the AS-Path is limited to a maximum of 15 AS numbers.

Policy Examples

Translating an Access Profile to a Policy

```
entry  entry-5  {
```

```
    If  {
        nlri    22.16.0.0/14;
    }
    then  {
        permit;
    }
}
entry  entry-10  {
    if  {
        nlri   192.168.0.0/18 exact;
    }
    then  {
        permit;
    }
}
entry  entry-15  {
    if  {
        nlri   any/8;
    }
    then  {
        deny;
    }
}
entry  entry-20  {
    if  {
        nlri   10.10.0.0/18;
}
Then  {
    permit;
}
}
entry  entry-25  {
    if  {
        nlri   22.44.66.0/23  exact;
    }
    then  {
        deny;
    }
}
```

The policy above can be optimized by combining some of the if statements into a single expression.

The compact form of the policy looks like this:

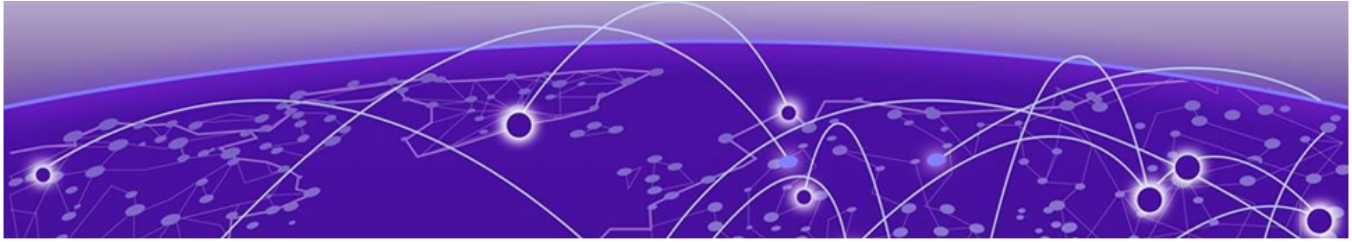
```
entry  permit_entry  {
    If match any {
        nlri    22.16.0.0/14;
        nlri   192.168.0.0/18 exact ;
        nlri   10.10.0.0/18;
    }
    then  {
        permit;
    }
}
entry  deny_entry  {
    if match any {
        nlri   any/8;
        nlri   22.44.66.0/23  exact;
    }
    then  {
        deny;
```

```
    }
}
```

Translating a Route Map to a Policy

```
entry  entry-10  {
    If  {
        origin    incomplete;
    }
    then  {
        permit;
    }
}
entry  entry-20  {
    if  {
        community   6553800;
    }
    then  {
        deny;
    }
}
entry  entry-30  {
    if  {
        med   30;
    }
    then  {
        next-hop     10.201.23.10;
        as-path 64502;
        as-path 64503;
        as-path     64504;
        as-path     64504;
        permit;
    }
}
entry  entry-40  {
    if  {
    }
    then  {
        local-preference 120;
        weight         2;
        permit;
}
}
entry  entry-50 match any {
    if  {
        origin         incomplete;
        community     19661200;
    }
    then  {
        dampening half-life 20 reuse-limit 1000 suppress-limit 3000 max-suppress 40
        permit;
    }
}
entry  entry-60 {
    if  {
        next-hop 192.168.1.5;
    }
    then  {
        community  add  949616660;
        permit;
```

```
    }
}
entry  deny_rest {
    if  {
    }
    then  {
        deny;
    }
}
```



Security

Security Features Overview on page 1531

Security Features Overview

General

Security is a term that covers several different aspects of network use and operation.

One general type of security is control of the devices or users that can access the network. Ways of doing this include authenticating the user at the point of logging in, controlling access by defining limits on certain types of traffic, or protecting the operation of the switch itself. Security measures in this last category include routing policies that can limit the visibility of parts of the network or denial of service protection that prevents the CPU from being overloaded. Another general type of security is data integrity and confidentiality, which is provided by the MACsec protocol. Finally, management functions for the switch can be protected from unauthorized use. This type of protection uses various types of user authentication.

Security Modes

For information on ExtremeXOS security modes, see Security Mode Overview on page 1533.

Security Features

ExtremeXOS has enhanced security features designed to protect, rapidly detect, and correct anomalies in your network. Extreme Networks products incorporate a number of features designed to enhance the security of your network while resolving issues with minimal network disruption. No one feature can ensure security, but by using a number of features in concert, you can substantially improve the security of your network.

The following list provides a brief overview of some of the available security features:

- *ACL (Access Control List)*s—ACLs are policy files used by the ACL application to perform packet filtering and forwarding decisions on incoming traffic and packets. Each packet arriving on an ingress port is compared to the ACL applied to that port and is either permitted or denied.

For more information about using ACLs to control and limit network access, see ACLs on page 341.

- CLEAR-Flow—CLEAR-Flow inspects Layer 2 and Layer 3 packets, isolates suspicious traffic, and enforces policy-based mitigation actions. Policy-based mitigation actions include the switch taking an immediate, predetermined action or sending a copy of the traffic off-switch for analysis.

  For more information about CLEAR-Flow, see CLEAR-Flow.
- Denial of Service Protection—DoS protection is a dynamic response mechanism used by the switch to prevent critical network or computing resources from being overwhelmed and rendered inoperative. In essence, DoS protection protects the switch, CPU, and memory from attacks and attempts to characterize the attack (or problem) and filter out the offending traffic so that other functions can continue. If the switch determines it is under attack, the switch reviews the packets in the input buffer and assembles ACLs that automatically stop the offending packets from reaching the CPU. For increased security, you can enable DoS protection and establish CLEAR-Flow rules at the same time.

  For more information about DoS attacks and DoS protection, see Denial of Service Protection on page 1586.
- Network Login—Controls the admission of user packets and access rights thereby preventing unauthorized access to the network. Network login is controlled on a per port basis. When network login is enabled on a port in a *VLAN (Virtual LAN)*, that port does not forward any packets until authentication takes place. Network login is capable of three types of authentication: web-based, MAC-based, and 802.1X.

  For more information about network login, see Network Login on page 1214.
- Policy Files—Text files that contain a series of rule entries describing match conditions and actions to take. Policy files are used by both routing protocol applications (routing policies) and the ACL application (ACLs).

  For more information about policy files, see Routing Policies on page 1520.
- Routing Policies—Policy files used by routing protocol applications to control the advertisement, reception, and use of routing information by the switch. By using policies, a set of routes can be selectively permitted or denied based on their attributes for advertisements in the routing domain. Routing policies can be used to “hide” entire networks or to trust only specific sources for routes or ranges of routes.

  For more information about using routing policies to control and limit network access, see .
- sFlow—A technology designed to monitor network traffic by using a statistical sampling of packets received on each port. sFlow also uses IP headers to gather information about the network. By gathering statistics about the network, sFlow becomes an early warning system, notifying you when there is a spike in traffic activity. Upon analysis, common response mechanisms include applying an ACL, changing *QoS (Quality of Service)* parameters, or modifying VLAN settings.

For more information, see Using sFlow on page 1770.

- MAC Security (MACsec)—A protocol designed to provide data integrity (ensure data has not been altered in an unauthorized manner) and data confidentiality (ensure data cannot be read by an unauthorized party). This feature provides line rate data encryption/decryption by the use of specialized cryptographic hardware.

  For more information about MACsec, see MAC Security with Pre-shared Key Authentication on page 1556.

Security Mode Overview

ExtremeXOS has multiple security modes used in multiple applications such as AAA, SSH, *SNMP (Simple Network Management Protocol)*, etc.:

- Secure mode (see Secure Mode on page 1533).
- FIPS mode (see Federal Information Processing Standards (FIPS) Mode on page 1533).
- Default normal mode.

Secure Mode

Secure mode only works in SSH. As per this mode, only strong ciphers and Message Authentication Codes (MACs) are allowed in SSH server and SSH client connections. After enabling secure-mode:

- For communication, SSH server uses a new secure-mode list made each for ciphers and MACs.
- For the SSH client, EPM is notified to change the bit dedicated to SSH secure-mode, which hides the weak ciphers and MACs from SSH client CLI commands.

Applications

Only SSH is uses this mode.

Federal Information Processing Standards (FIPS) Mode

Federal Information Processing Standards (FIPS) is a collection of standards defined by the United States Federal Government for document process, encryption algorithms, personal identification and verification, as well as many other areas. These standards are used within the United States federal government (civilian agencies, Department of Defense, and intelligence agencies).

FIPS, in the context of ExtremeXOS, refers to the specific standard 140-2, *System Requirements for Cryptographic Modules*. The publication series was issued to coordinate the requirements and standards for cryptographic modules that include both hardware and software components. Protection of a cryptographic module within a security system is necessary to maintain the confidentiality and integrity of the information protected by the module. This standard specifies the security requirements that are satisfied by a cryptographic module. The standard provides four increasing, qualitative levels of security intended to cover a wide range of potential applications and environments. The security requirements cover areas related to the secure design and implementation of a cryptographic module.

These areas include:

- Cryptographic module specification
- Cryptographic module ports and interfaces
- Roles, services, and authentication
- Finite state model; physical security
- Operational environment
- Cryptographic key management
- Electromagnetic interference/electromagnetic compatibility (EMI/EMC)
- Self-tests
- Design assurance
- Mitigation of other attacks

The US Government requirements break down FIPS 140-2 into two distinct categories: FIPS 140-2 Certified and FIPS 140-2 Compliant. Compliant is needed when only network management data is encrypted or decrypted, which typically includes SSH, SNMPv3, etc. When the product encrypts and decrypts user data this changes the requirement to FIPS 140-2 Certified. An example of a product that would require FIPS 140-2 certification is wireless, since user data is encrypted and decrypted as it flows across the radio waves.

In ExtremeXOS for all cryptographic functions, applications generally use OpenSSL library. OpenSSL module itself is not validated, so a new carefully defined software component called the OpenSSL FIPS Object Module has been created. It is designed for compatibility with the OpenSSL library so products using the OpenSSL library and API can be converted to use FIPS 140-2 validated cryptography with minimal effort. The most recent open source-based validation is the OpenSSL FIPS Object Module v2.0, FIPS 140-2 certificate #1747.

ExtremeXOS uses both OpenSSL libraries:

- openssl-1.0.2l—When FIPS Mode is off.
- openssl-fips-2.0.16—When FIPS Mode is on.

Note
Digital Signature Algorithm (DSA) is not supported in FIPS mode. After turning on FIPS mode, you need to generate a new SSH host key.

Applications

SSH, *SNMP*, AAA, NTP, and EMS use this mode.

Ciphers and Message Authentication Codes (MACs) Supported by Security Modes

Cryptographic Functions	Full Name (Default Mode)	FIPS Mode	Secure Mode (SSH Only)
Hash Function (used for authentication)			
In SNMPv3	*MD5 (Message-Digest algorithm 5)*		
	SHA	Yes	
	HMAC-MD5		
	HMAC-SHA-1	Yes	
In SSH	hmac-md5-96		
	hmac-md5-etm@openssh.com		
	hmac-md5-96-etm@openssh.com		
	hmac-sha1	Yes	Yes
	hmac-sha1-96		
	hmac-sha1-etm@openssh.com		
	hmac-sha1-96-etm@openssh.com		
	hmac-sha2-256	Yes	Yes
	hmac-sha2-512	Yes	Yes
	hmac-sha2-256-etm@openssh.com		
	hmac-sha2-512-etm@openssh.com		
In AAA	SHA-256	Yes	Yes
In NTP	SHA-256	Yes	
Symmetric Algorithms (used for encryption of file or text)			
In SSH, SNMPv3	DES		
	AES128	Yes	
	AES192	Yes	
	AES256	Yes	
In SSH	AES128-CTR	Yes	Yes
	AES192-CTR	Yes	Yes
	AES256-CTR	Yes	Yes
	rijndael-cbc@lysator.liu.se	Yes	
	chacha20-poly1305@openssh.com		
Asymmetric Algorithms (Factorization of integers)			
In SSH	DSA (Authentication and Digital Signature)	Yes	

Cryptographic Functions	Full Name (Default Mode)	FIPS Mode	Secure Mode (SSH Only)
	RSA (Authentication and Encryption)	Non-approved, but allowed for Key Encryption Decryption	
	diffie-hellman-group1-sha1 exchange key (authentication)	Non-approved, but allowed for key exchange	

Priority with Multiple Security Modes

Multiple modes can be in effect at the same time for certain applications (for example, all modes are applicable for SSH).

The precedence from highest priority to the lowest:

1. Secure mode
2. FIPS mode
3. Default mode

Example: If for SSH, FIPS mode and Secure Mode are both enabled, Secure mode takes precedence over FIPS mode.

Using Safe Defaults Mode

When you take your switch from the box and set it up for the first time, you set the safe defaults mode. You should use the safe defaults mode, which lets you configure IP management connectivity if you choose not to use Auto Provisioning It also lets you choose whether to enable or disable Telnet, *STP (Spanning Tree Protocol)*, and *SNMP*. All ports are enabled in the factory default setting; you can choose to have all unconfigured ports disabled on reboot using the interactive questions. Also, *STPD (Spanning Tree Domain)* s0 is enabled on the default *VLAN*; you have the option to disable STPD in safe defaults mode.

After you connect to the console port of the switch, or after you run `unconfigure switch {all}` or `run provisioning`, you can change management access to your device to enhance security.

1. Connect the console and log in to the switch.

```
This switch currently has some management methods enabled for convenience reasons.
Please answer these questions about the security settings you would like to use.
You may quit and accept the default settings by entering 'q' at any time.

Would you like to change the switch OS to Fabric Engine? [y/N/q]
```

2. You are prompted to select your network operating system: Switch Engine (default) or Fabric Engine:
   - For Switch Engine, type `N`.
   - For Fabric Engine, type `y`.

3. Type `y` (to disable) or `n` (to enable) auto-provisioning.

```
By default, Auto-Provisioning uses DHCP on all Ethernet ports as this switch
attempts to connect to an Extreme Networks management product.
Instead of using DHCP, do you want to 'disable auto-provision' and
configure a static IP address, default gateway and DNS server now? [y/N/q]: y
```

   Select `y` to be prompted for a port to use for management, an IP address and subnet mask length, an optional default gateway IP address, a DNS name server and DNS domain suffix.

4. Type `y` (to disable) or `n` (to enable) *MSTP (Multiple Spanning Tree Protocol)*.

```
The switch offers an enhanced security mode. Would you like to read more,
and have the choice to enable this enhanced security mode? [y/N/q]:
```

   If you select "no," go to 6 on page 1537.

5. If you select "yes," the following appears:

```
Enhanced security mode configures the following defaults:

        * Disable Telnet server.
        * Disable HTTP server.
        * Disable SNMP server.
        * Remove all factory default login accounts.
        * Force creation of a new admin (read-write) account.
        * Lockout accounts for 5 minutes after 3 consecutive login failures.
        * Plaintext password entry will not be allowed.
        * Generate an event when the logging memory buffer exceeds 90% of capacity.
        * Only admin privilege accounts are permitted to run "show log".
        * Only admin privilege accounts are permitted to run "show diagnostics".

Would you like to use this enhanced security mode? [Y/n/q]:
```

   If you select "yes," enhanced security mode is enabled. Go to step 10 on page 1538.

6. If you select "no," you are prompted to disable Telnet:

```
Telnet is enabled by default. Telnet is unencrypted and has been the target of
security exploits in the past.

Would you like to disable Telnet? [y/N/q]:
```

7. You are prompted to enable SNMPv2c:

```
SNMP access is disabled by default.
SNMPv2c uses no encryption, SNMPv3 can be configured to eliminate this problem.
Would you like to enable SNMPv2c? [y/N/q]: Yes
```

8. You are prompted to set up the community string:

```
SNMP community string is a text string that is used to authenticate SNMPv2c messages.
It is required for managing the switch using SNMPv2c.
Would you like to configure a read-only and read-write community string? [Y/n/q]: Yes

Read-only community string:
Re-enter read-only community string:
Read-write community string:
Re-enter read-write community string:
```

9. You are prompted to enable SNMPv3:

```
Would you like to enable SNMPv3? [y/N/q]: Yes

SNMPv3 uses usernames/passwords to authenticate and encrypt SNMP messages.
Would you like to create an SNMPv3 user? [Y/n/q]: Yes
```

```
User name: admin
Authentication password:
Reenter authentication password:
Privacy password:
Reenter privacy password:

SNMPv3 user 'admin' was created with authentication protocol SHA and privacy protocol
AES-128.
```

10. Reboot the switch.

MAC Security

The switch maintains a database of all media access control (MAC) addresses received on all of its ports.

The switch uses the information in this database to decide whether a frame should be forwarded or filtered. MAC security (formerly known as MAC address security) allows you to control the way the *FDB (forwarding database)* is learned and populated. For more information, see FDB.

MAC security includes several types of control. You can:

- Limit the number of dynamically learned MAC addresses allowed per virtual port. For more information, see Limiting Dynamic MAC Addresses on page 1548.
- "Lock" the FDB entries for a virtual port, so that the current entries will not change, and no additional addresses can be learned on the port. For information, see MAC Address Lockdown on page 1551.

  Note
  You can either limit dynamic MAC FDB entries or lockdown the current MAC FDB entries, but not both.

- Set a timer on the learned addresses that limits the length of time the learned addresses will be maintained if the devices are disconnected or become inactive. For more information, see MAC Address Lockdown with Timeout on page 1551.

  Note
  When limit-learning is configured in the port which is also associated with some other vlan where learning is disabled, then few packets with new MAC address beyond learning limit will get flooded. This flooding will take place for fraction of second until new black-hole entry is created in hardware.

- Use ACLS to prioritize or stop packet flows based on the source MAC address of the ingress virtual LAN (*VLAN*) or the destination MAC address of the egress VLAN. For more information about *ACL* policies, see Security on page 1628.
- Enhance security, depending on your network configuration, by disabling Layer 2 flooding. For more information about enabling and disabling Layer 2 flooding, see Managing Egress Flooding on page 833.

MAC Locking

MAC locking helps prevent unauthorized access to the network by limiting access based on a device's MAC address. MAC locking enables the binding of specific MAC

addresses to specific ports on a switch. MAC locking locks a port to one or more MAC addresses, preventing connection of unauthorized devices via a port. With MAC locking enabled, the only frames forwarded on a MAC locked port are those with the configured or dynamically selected MAC addresses for that port.

Frames received on a port with a Source MAC address not bound to the port are discarded or optionally allowed to dynamically bind to the port, up to a user-controlled maximum number of MAC addresses per port.

There are two different types of MAC locking:

• Static MAC locking - Locking one or more specified MAC addresses to a port.

• First Arrival MAC locking - Locking one or more MAC addresses to a port based on first arrival of received frames after First Arrival MAC locking is enabled. The configuration specifies the maximum number of end users that will be allowed. As each new end user is identified, it is MAC locked up to the maximum number of users. Once the maximum number of users has been MAC locked, all other users will be denied access to the port until a MAC locked address is either aged, if aging is configured, or the session for that user ends.

The MAC locking feature is disabled in the device, by default. MAC locking must be enabled both globally and on port level. Once enabled, ports can be configured for static and First Arrival MAC locking.

Existing limit learning and lock learning features are supported on a port-*VLAN* combination. The MAC locking feature implemented in ExtremeXOS 15.7 supports MAC locking functionality on a port basis.

MAC Locking Limitations

The following limitations exist for the MAC locking feature:

- MAC locking cannot be enabled along with existing limit learning and lock learning features.
- MAC locking is be supported on ports in the VPLS service *VLAN*.

MAC Locking Functionality

This feature provides a global enable/disable control and controls per port. Additional controls are provided to create, destroy, enable and disable static MAC to Interface bindings. The MAC and port, together are required to be unique (the same MAC may be bound to multiple ports).

The configuration of a maximum number of static and dynamic entries (each individually) is provided per port. First Arrival bindings (also known as first arrival MAC locking stations) are not persistent through a device reset. Static bindings (also known as static MAC locking stations) are maintained through a device reset. First Arrival bindings are removed in the event of link loss, or after the *FDB* entry with the MAC locked MAC ages out. A port, may have both the static and dynamic entries, at any instance.

Dynamic locking may be disabled by setting the maximum number of First Arrival MAC addresses to zero.

Controls are provided per port to convert all current first arrival entries to static entries. This converts only the first arrival MAC bindings to static bindings. This is not to be confused with MAC locking where all dynamic FDB entries are converted to static permanent locked FDB entries and further learning is disabled. Controls are provided to clear current (both static and dynamic) bindings. Each port is configurable to support the aging of dynamically locked bindings.

The device keeps a record of the number of MAC locked stations and when the configured threshold is reached, a threshold *SNMP* trap/notification and/or a log message is issued based on the per-port configuration. The Source MAC Address of the frame causing the last invalid attempt is also recorded. In the event that the device is so configured, a violation SNMP Trap/ Notification and/or a violation log message is issued – these controls will be exercised per port.

Configuring MAC Locking

The following content describes the MAC locking commands:

Enabling and Disabling MAC Locking Globally

To enable or disable the MAC locking feature, use the following commands:

- `enable mac-locking`
- `disable mac-locking`

Enabling and Disabling MAC Locking on Ports

To enable MAC locking on specific ports, use the following command:

- `enable mac-locking ports [`**`all`**`| port_list]`

To Disable MAC locking on specific ports, use the following command:

- `disable mac-locking ports [`**`all`**`| port_list]`

Configuring First-arrival/Dynamic MAC Locking Limit

To configure first arrival MAC locking on a port, use the following command:

- `configure mac-locking ports port_list first-arrival limit-learning learn_limit`

When the configured limit is reached, no further entries are learnt. However, if the learnt entries are aged out, new MAC addresses can be learned.

By default, Aging is disabled for first arrival MAC locking entries on a configured port. When the *FDB* entries are aged out, they are removed from the FDB, but they are retained in the MAC locking table. So when the first arrival limit is reached, only those entries in the MAC locking table can be learned again if these devices start sending out traffic. Any new MAC addresses cannot be learned.

The maximum number of dynamic MAC addresses that can be locked on a port is 600.

Note
There is no command to unconfigure first arrival or static MAC locking limit. The value can be reset giving the default learn limit specified in the help text. When First arrival MAC locking is configured to a value that is lower than the number of MACs that are locked, all the MAC locking bindings on the port are cleared.

Configuring Static MAC Locking Limit

To configure static MAC locking on a port, use the following command:

- `configure mac-locking ports port_list static limit-learning learn_limit`

When the configured limit is reached, no further entries are learned (either black holed or not further learned depending on the configured action). However, if the learned entries are cleared or deleted, new MAC addresses can be created and learned. The maximum number of MAC addresses that will be locked on a port configured for static MAC locking is 64. Aging is not applicable for the static MAC locking entries.

Note
There is no command to unconfigure first arrival or static MAC locking limit. The value can be reset giving the default learn limit specified in the help text. CLI doesn't allow changing the static MAC locking limit value to a value lower than the number of MACs locked in the MAC lock station table. Some or all of the created MAC locking stations should be removed to change the limit to a lower value.

Note
Assume that port 2:22 is enabled for MAC Locking. The maximum static entry limit value is configured to 5 on port 2:22. If the user wants to configure the maximum static entry limit value to 3,

1. The device will display an error, as maximum number of static stations already locked on the port and the value cannot be reduced.
2. In the same example, the port 2:22 has only 1 static station locked. If the maximum static entry limit value is reduced to 3, the device will allow to reduce the value.

Scenario A:

```
* Slot-1 DUT1.94 # show mac-locking ports 2:22

MAC locking is globally enabled.
Port   MAC  Trap      Log       FA    Limit     Link    Max Max   Last Violating
       Lock Thr|Viol Thr|Viol Aging   Action    Down    Stc FA    MAC Address
       Stat                         Cfg|Stat   Action
-----  ---- -------- -------- ----- --------    ------ --- ---   -----------------
2:22   ena  off|on    off|on   dis   ena|ena   clear    5   45   00:11:11:11:11:04

Legend:
```

```
Stat              - Status                    Thr|Viol - Threshold | Violation

Max Stc           - Max Static Count          Max FA   - Max First-Arrival Count

dis               - Disabled                  ena       - Enabled

retain            - Retain MACs               clear     - Clear MACs

Limit Action Cfg  - If port should be disabled when learnt limit is exceeded

             dis   - Port to be disabled when learn limit is exceeded

             ena   - Port to remain enabled when learn limit is exceeded

Limit Action Stat - Port status on exceeding learn limit

*  Slot-1 DUT1.95 #

*  Slot-1 DUT1.95 # show mac-locking stations ports 2:22

Port   MAC Address         Status     State           Aging

-----  -----------------   --------   -------------   -----

2:22   00:11:11:11:11:00   active     static          false

2:22   00:11:11:11:11:01   active     static          false

2:22   00:11:11:11:11:02   active     static          false

2:22   00:11:11:11:11:03   active     static          false

2:22   00:11:11:11:11:04   active     static          false

*  Slot-1 DUT1.96 # configure mac-locking ports 2:22 static limit-learning 3

Error: Static limit-learning value cannot be reduced to 3 for port 2:22 as 5 static
stations are already created.


Configuration failed on backup Node, command execution aborted!

* Slot-1 DUT1.97 #
```

Scenario B:

```
* Slot-1 DUT1.109 # show mac-locking stations ports 2:22

Port   MAC Address         Status     State           Aging

-----  -----------------   --------   -------------   -----

2:22   00:11:11:11:11:00   active     static          false

2:22   00:11:11:11:11:01   active     first-arrival   false

2:22   00:11:11:11:11:02   active     first-arrival   false

2:22   00:11:11:11:11:03   active     first-arrival   false

2:22   00:11:11:11:11:04   active     first-arrival   false
```

```
* Slot-1 DUT1.109 # show mac-locking ports 2:22

MAC locking is globally enabled.

Port   MAC  Trap      Log       FA    Limit     Link    Max Max   Last Violating

       Lock Thr|Viol Thr|Viol Aging   Action    Down    Stc FA    MAC Address

       Stat                          Cfg|Stat   Action

-----  ---- --------  --------  ----- --------  ------  --- ---   -----------------

2:22   ena  off|on    off|on    dis   ena|ena   clear     5   45   00:11:11:11:11:04

Legend:

Stat              - Status                    Thr|Viol - Threshold | Violation

Max Stc           - Max Static Count          Max FA   - Max First-Arrival Count

dis               - Disabled                  ena       - Enabled

retain            - Retain MACs               clear     - Clear MACs

Limit Action Cfg  - If port should be disabled when learnt limit is exceeded

             dis   - Port to be disabled when learn limit is exceeded
             ena   - Port to remain enabled when learn limit is exceeded

Limit Action Stat - Port status on exceeding learn limit

* Slot-1 DUT1.110 # configure mac-locking ports 2:22 static limit-learning 3

* Slot-1 DUT1.111 # show mac-locking ports 2:22

MAC locking is globally enabled.

Port   MAC  Trap      Log       FA    Limit     Link    Max Max   Last Violating

       Lock Thr|Viol Thr|Viol Aging   Action    Down    Stc FA    MAC Address

       Stat                          Cfg|Stat Action

-----  ---- --------  --------  ----- --------  ------  --- ---   -----------------

2:22   ena  off|on    off|on    dis   ena|ena   clear     3   45   00:11:11:11:11:04

Legend:

Stat              - Status                    Thr|Viol - Threshold | Violation

Max Stc           - Max Static Count          Max FA   - Max First-Arrival Count

dis               - Disabled                  ena       - Enabled

retain            - Retain MACs               clear     - Clear MACs

Limit Action Cfg  - If port should be disabled when learned limit is exceeded

             dis   - Port to be disabled when learn limit is exceeded
             ena   - Port to remain enabled when learn limit is exceeded

Limit Action Stat - Port status on exceeding learn limit
```

Create/Enable/Disable static MAC Locking Entries

To create a static MAC locking entry (also known as MAC locking station) and enable or disable MAC locking for the specific MAC address and port, use the following command:

```
configure mac-locking ports port_list static [add | enable | disable]
station station_mac_address
```

Note
A static MAC locking station is enabled by default.

To disable the static MAC locking station, use the following command.

```
configure mac-locking ports port_list static disable station
station_mac_address
```

When created and enabled, a static MAC lock configuration allows only the end station designated by the MAC address to participate in forwarding of traffic.

The disabled entries are also counted when calculating the total number of locked stations. Static MAC locking stations that are disabled are only shown in “show mac-locking stations static” command. When “static” keyword is not given in “show mac-locking stations”, the disabled entries are not shown.

Enable/Disable Aging of First-arrival MAC Addresses

To enable or disable first arrival MAC address aging, use the following command.

```
configure mac-locking ports port_list first-arrival aging [enable |
disable]
```

Dynamic MAC locking mode MAC address aging is disabled by default.

This is applicable only to MAC addresses locked by first-arrival locking and not to MAC addresses locked by static locking.

When First arrival aging is disabled, MAC locking stations are retained even when the corresponding FDB entry ages out.

When First arrival aging is enabled, MAC locking station starts aging when all the FDB entries corresponding to the station MAC are removed. MAC lock stations do not start aging when FDB entries are present.

When “first arrival aging” is configured to be enabled in first-arrival locking, when an FDB entry ages out, the entry is no more locked and so new MAC addresses can be learned till the configured first-arrival limit is reached.

Note
First arrival Aging – Age out time for First Arrival MAC locking station is same as FDB aging time that is configured using `configure fdb agingtime`.

Move First-arrival MACs to Static Entries

To move all current first-arrival MACs to static entries on a port, use the following command:

```
configure mac-locking ports port_list first-arrival move-to-static
```

This command converts dynamic MAC locked stations to static MAC locked stations. There is no change to FDB entries.

The static MAC locked station entries are saved in configuration and so are preserved across reboots.

Note
Ensure the static limit can accommodate the entries before moving them from to static. Otherwise, the device may throw the following error: `Error: Some dynamic stations could not be converted to static stations for port <port_list>`.

Note
An FDB entry created from the CLI will not be removed when a static MAC lock station is created and disabled for the corresponding MAC address. It is necessary to delete the FDB entry from the CLI. MAC-Locking does not remove user created FDB entries.

Managing MAC Locking

Enable/Disable Clearing First-arrival MACs on Link Change

To manage the behavior of first arrival MAC locking on link state change, use the following command.

```
configure mac-locking ports port_list first-arrival link-down-action
[clear-macs | retain-macs]
```

Clear MAC on link change is enabled by default.

When the link goes down, by default, all the first arrival MAC locking addresses will be removed. When link-down-action is configured to “retain-macs”, the first arrival MAC locking addresses will be retained even when the link goes down.

Disable/Enable port when MAC threshold is reached

This command is used to configure the disabling of ports when the configured MAC threshold is met. This is used for both “first arrival” and “static” MAC locking methods.

```
configure mac-locking ports port_list learn-limit-action [disable-port |
remain-enabled
```

The port is disabled when the configured MAC threshold is met. All the *FDB* entries learned on this port are flushed as the port is disabled. This configuration can be reset using the `clear mac-locking disabled-state ports port_list` command. When MAC locking is disabled on the port, the port comes back up.

Clearing the Disabled-state of a Port

This command is used to return the behavior of first arrival MAC locking with link state change to its default value of enabled.

```
clear mac-locking disabled-state ports port_list
```

Delete Static MAC Locking Entries

To delete MAC locking for all static MAC address or the specified static MAC address on the given port, use the following command:

```
configure mac-locking ports port_list static delete station
[station_mac_address | all]
```

Clearing MAC Locking entries

The following command is used to clear MAC locking station entries for the given parameters:

```
clear mac-locking station [all | {mac station_mac_address} {first-
arrival | static} {ports port_list}]
```

This command clears MAC locking configuration by port/mac/first arrival/static etc.

Note
Clearing static MAC locking stations will remove them from the configuration. The cleared static MAC locking stations will not be saved across reboots.

Displaying MAC Locking Information

This command is used to display the status of MAC locking on one or more ports.

```
show mac-locking {ports port_list}
```

Note
In MLAG, the mac-locking entries shown in this command's output are only natively learned FDB entries on the switch.

If port is not specified, MAC locking status will be displayed for all ports.

Sample output:

```
Slot-1 Stack.2 # show mac-locking

MAC locking is globally enabled.

Port   MAC  Trap      Log       FA    Limit     Link    Max Max   Last Violating
    Lock Thr|Viol    Thr|Viol   Aging Action    Down    Stc FA    MAC Address
   Stat                          Cfg|Stat Action
-----  ---- -------- -------- ----- --------    ------ --- ---    -----------------
1:1    dis  off|off   off|off   dis  ena|ena   clear    64 600    00:00:00:00:00:00
1:2    dis  off|off   off|off   dis  ena|ena   clear    64 600    00:00:00:00:00:00
1:3    dis  off|off   off|off   dis  ena|ena   clear    64 600    00:00:00:00:00:00
1:4    dis  off|off   off|off   dis  ena|ena   clear    64 600    00:00:00:00:00:00
1:5    dis  off|off   off|off   dis  ena|ena   clear    64 600    00:00:00:00:00:00

Legend:
Stat              - Status                   Thr|Viol - Threshold | Violation
```

```
Max Stc           - Max Static Count          Max FA   - Max First-Arrival Count
dis               - Disabled                  ena       - Enabled
retain            - Retain MACs               clear     - Clear MACs
Limit Action Cfg  - If port should be disabled when learnt limit is exceeded
             dis   - Port to be disabled when learn limit is exceeded
             ena   - Port to remain enabled when learn limit is exceeded
Limit Action Stat - Port status on exceeding learn limit
```

The following command displays MAC locking stations for different parameters:

```
show mac-locking stations {first-arrival | static} {ports port_list}
```

MAC Locking Configuration Example

The following command enables MAC locking both globally for the device (stack) and at the port level for ports 2:1 through 2: 5:

```
Slot-1 Stack.1 # enable mac-locking

Slot-1 Stack.2 # enable mac-locking ports 2:1-2:5
```

The following command lines enable port 2:1 for a maximum of 3 static MAC address entries. This is followed by four static MAC address creation entries. The fourth entry fails because the maximum allowed has been set to 3.

```
* Slot-1 Stack.3 # configure mac-locking ports 2:1 static limit-learning 3
* Slot-1 Stack.4 # configure mac-locking ports 2:1 static add station 00:22:33:44:55:66
* Slot-1 Stack.5 # configure mac-locking ports 2:1 static add station 00:22:33:44:55:77
* Slot-1 Stack.6 # configure mac-locking ports 2:1 static add station 00:22:33:44:55:88
* Slot-1 Stack.7 # configure mac-locking ports 2:1 static add station 00:22:33:44:55:99
Error: Station 00:22:33:44:55:99 cannot be added as maximum static limit of 3 is already
reached on port 2:1.


* Slot-1 Stack.10 # show mac-locking stations static ports 2:1
Port   MAC Address         Status     State           Aging
-----  -----------------   --------   -------------   -----
2:1    00:22:33:44:55:66   active     static          false
2:1    00:22:33:44:55:77   active     static          false
2:1    00:22:33:44:55:88   active     static          false

Total for specified ports: 3 Static: 3 First-Arrival: 0
* Slot-1 Stack.11 #
```

The following commands configure ports 2:2 through 2:5 for dynamic MAC locking with a maximum of 15 users on each port. This is followed by a line enabling MAC locking trap messaging on ports 2:1 through 5:

```
* Slot-1 Stack.12 # configure mac-locking ports 2:2-2:5 first-arrival limit-learning 15
* Slot-1 Stack.13 # configure mac-locking ports 2:2-2:5 trap on
* Slot-1 Stack.14 # show mac-locking ports 2:1-2:5

MAC locking is globally enabled.

Port   MAC  Trap      Log       FA    Limit     Link    Max Max   Last Violating
       Lock Thr|Viol Thr|Viol Aging   Action    Down    Stc FA    MAC Address
       Stat                         Cfg|Stat   Action
-----  ---- -------- -------- ----- --------    ------ --- ---    -----------------
2:1    ena  off|off   off|off   dis  ena|ena   clear     3 600    00:00:00:00:00:00
2:2    ena  off|on    off|off   dis  ena|ena   clear    64   15   00:00:00:00:00:00
```

```
2:3    ena  off|on    off|off   dis  ena|ena   clear    64   15   00:00:00:00:00:00
2:4    ena  off|on    off|off   dis  ena|ena   clear    64   15   00:00:00:00:00:00
2:5    ena  off|on    off|off   dis  ena|ena   clear    64   15   00:00:00:00:00:00

Legend:
Stat              - Status                    Thr|Viol - Threshold | Violation
Max Stc           - Max Static Count          Max FA   - Max First-Arrival Count
dis               - Disabled                  ena      - Enabled
retain            - Retain MACs               clear     - Clear MACs
Limit Action Cfg  - If port should be disabled when learnt limit is exceeded
             dis   - Port to be disabled when learn limit is exceeded
             ena   - Port to remain enabled when learn limit is exceeded
Limit Action Stat - Port status on exceeding learn limit
* Slot-1 Stack.15 #
```

Limiting Dynamic MAC Addresses

You can set a predefined limit on the number of dynamic MAC addresses that can participate in the network.

After the *FDB* reaches the MAC limit, all new source MAC addresses are blackholed at both the ingress and egress points. These dynamic blackhole entries prevent the MAC addresses from learning and responding to *ICMP (Internet Control Message Protocol)* and address resolution protocol (ARP) packets.

Note

Blackhole FDB entries added due to MAC security violations are removed after each FDB aging period regardless of whether the MAC addresses in question are still sending traffic. If the MAC addresses are still sending traffic, the blackhole entries will be re-added after they have been deleted.

Configuring Limit Learning

- To limit the number of dynamic MAC addresses that can participate in the network, use the limit-learning option in following command:

  `configure ports` *port_list* {**tagged** *tag*} **vlan** *vlan_name* | *vlan_list* [**limit-learning** *number* {**action** [**blackhole** | **stop-learning**]} | **lock-learning** | **unlimited-learning** | **unlock-learning**]

  This command specifies the number of dynamically learned MAC entries allowed for these ports in this *VLAN*. The range is 0 to 500,000 addresses.

  When the learned limit is reached, all new source MAC addresses are blackholed at the ingress and egress points. This prevents these MAC addresses from learning and responding to *ICMP* and ARP packets.

  Dynamically learned entries still get aged and can be cleared. If entries are cleared or aged out after the learning limit has been reached, new entries will then be able to be learned until the limit is reached again.

  Permanent static and permanent dynamic entries can still be added and deleted using the `create fdb` and `disable flooding ports` commands. These override any dynamically learned entries.

For ports that have a learning limit in place, the following traffic still flows to the port:

- Packets destined for permanent MAC addresses and other non-blackholed MAC addresses
- Broadcast traffic
- *EDP (Extreme Discovery Protocol)* traffic

Traffic from the permanent MAC and any other non-blackholed MAC addresses still flows from the virtual port.

- To remove the learning limit, use the **unlimited-learning** option.

  `configure ports` *port_list* {**tagged** *tag*} **vlan** *vlan_name* | *vlan_list* [**limit-learning** *number* {**action** [**blackhole** | **stop-learning**]} | **lock-learning** | **unlimited-learning** | **unlock-learning**]

  The MAC limit-learning feature includes a stop-learning argument that protects the switch from exhausting *FDB* resources with blackhole entries. When limit-learning is configured with stop-learning, the switch is protected from exhausting FDB resources by not creating blackhole entries. Any additional learning and forwarding is prevented, but packet forwarding is not impacted for existing FDB entries.

Display Limit Learning Information

- To verify the configuration, enter the commands:

  `show vlan` *vlan name* **security**

  This command displays the MAC security information for the specified *VLAN*.

  `show ports` {**mgmt** | *portlist*} **info** {**detail**}

  This command displays detailed information, including MAC security information, for the specified port.

Example of Limit Learning

In Figure 188, three devices are connected through a hub to a single port on the Extreme Networks device.

If a learning limit of three is set for that port, and you connect a fourth device to the same port, the switch does not learn the MAC address of the new device; rather, the switch blackholes the address.

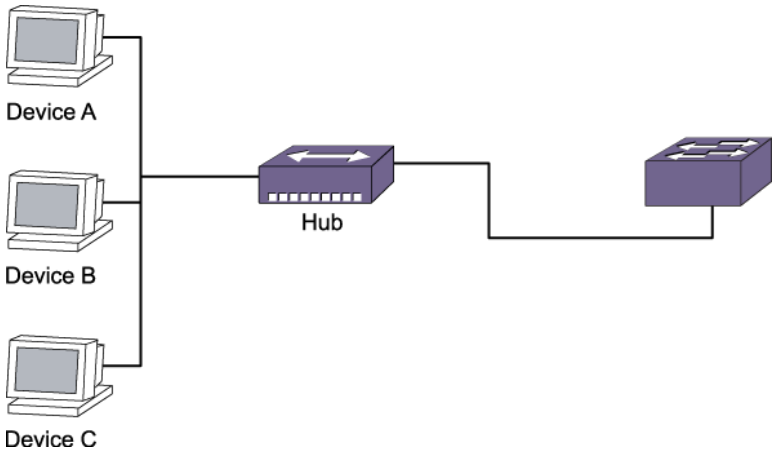


Figure 188: Switch Configured for Limit Learning

Limiting MAC Addresses with ESRP Enabled

If you configure a MAC address limit on *VLAN*s that participate in an *ESRP (Extreme Standby Router Protocol)* domain, you should add an additional back-to-back link (that has no MAC address limit on these ports) between the ESRP-enabled switches.

Doing so prevents ESRP protocol data units (PDUs) from being dropped due to MAC address limit settings.

Figure 189 is an example of configuring a MAC address limit on a VLAN participating in an ESRP domain.

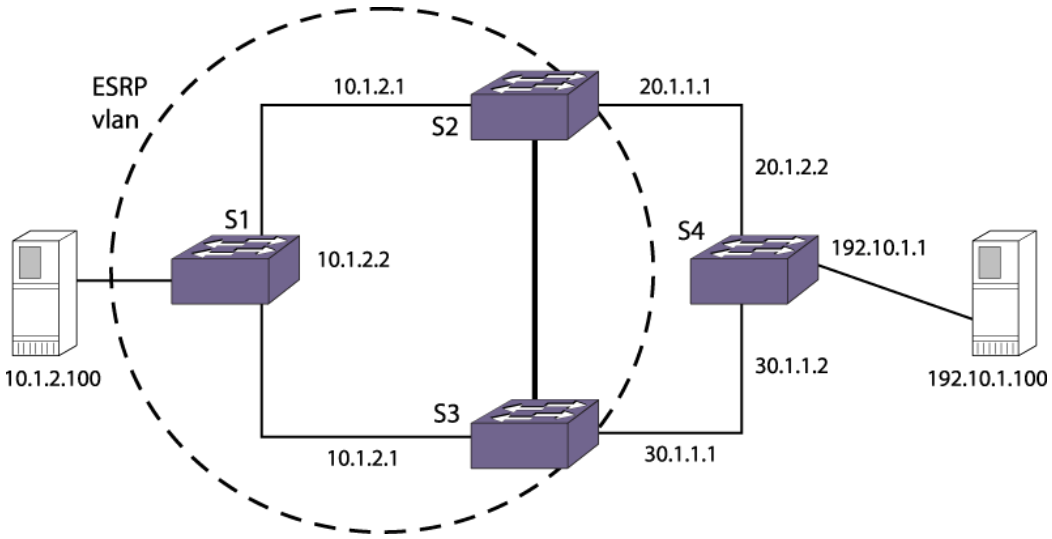


Figure 189: MAC Address Limits and VLANs Participating in ESRP

S2 and S3 are ESRP-enabled switches, while S1 is an ESRP-aware (regular Layer 2) switch. Configuring a MAC address limit on all S1 ports might prevent ESRP communication between S2 and S3. To resolve this, you should add a back-to-back link between S2 and S3. This link is not needed if MAC address limiting is configured only on S2 and S3, but not on S1.

MAC Address Lockdown

- In contrast to limiting learning on virtual ports, you can lockdown the existing dynamic *FDB* entries and prevent any additional learning using the lock-learning option from the following command:

  ```
  configure ports port_list {tagged tag} vlan vlan_name | vlan_list
  [limit-learning number {action [blackhole | stop-learning]} | lock-
  learning | unlimited-learning | unlock-learning]
  ```

  This command causes all dynamic FDB entries associated with the specified *VLAN* and ports to be converted to locked static entries. It also sets the learning limit to 0, so that no new entries can be learned. All new source MAC addresses are blackholed.

  Note
  Blackhole FDB entries added due to MAC security violations are removed after each FDB aging period regardless of whether the MAC addresses in question are still sending traffic. If the MAC addresses are still sending traffic, the blackhole entries will be re-added after they have been deleted.

  Locked entries do not get aged, but can be deleted like a regular permanent entry.

  For ports that have lock-down in effect, the following traffic still flows to the port:
  - Packets destined for the permanent MAC and other non-blackholed MAC addresses
  - Broadcast traffic
  - *EDP* traffic
  - Traffic from the permanent MAC still flows from the virtual port.
- Remove MAC address lockdown, use the unlock-learning option.

  ```
  configure ports port_list {tagged tag} vlan vlan_name | vlan_list
  [limit-learning number {action [blackhole | stop-learning]} | lock-
  learning | unlimited-learning | unlock-learning]
  ```

  When you remove the lockdown using the unlock-learning option, the learning-limit is reset to unlimited, and all associated entries in the FDB are flushed.
- Display the locked entries on the switch.

  ```
  show fdb
  ```

  Locked MAC address entries have the “l” flag.

MAC Address Lockdown with Timeout

The MAC address lockdown with timeout feature provides a timer for aging out MAC addresses on a per port basis and overrides the *FDB* aging time. That is, when this feature is enabled on a port and the port goes down/restarted, MAC addresses learned on that port age out based on the MAC lockdown timeout corresponding to the port, not based on the FDB aging time. By default, the MAC address lockdown timer is disabled.

Note
MAC address lockdown is not supported on NetLogin Dot1x non-policy mode.

When this feature is enabled on a port, MAC addresses learned on that port remain locked for the MAC lockdown timeout duration corresponding to the port, even when the port goes down. As a result, when a device is directly connected to the switch and then disconnected, the MAC address corresponding to the device will be locked up for the MAC lockdown timeout duration corresponding to that port. If the same device reconnects to the port before the MAC lockdown timer expires and sends traffic, the stored MAC address becomes active and the MAC lockdown timer is restarted. If the device is not reconnected for the MAC lockdown timeout duration, the MAC entry is removed.

MAC lockdown timeout entries are dynamically learned by the switch, which means these entries are not saved or restored during a switch reboot. If the switch reboots, the local MAC entry table is empty, and the switch needs to relearn the MAC addresses.

MAC address lockdown with timeout is configured by individual ports. The lockdown timer and address learning limits are configured separately for a port.

Note
You cannot enable the lockdown timeout feature on a port that already has MAC address lockdown enabled. For more information about MAC address lockdown, see MAC Address Lockdown on page 1551.

MAC address learning limits and the lockdown timer work together in the following ways:

- When the learning limit has been reached on a port, a new device attempting to connect to the port has its MAC address blackholed.
- As long as the timer is still running for a MAC entry, a new device cannot connect in place of the device that entry represents. That is, if a device has disconnected from a port, a new device cannot replace it until the lockdown timer for the first device has expired. This condition is true if the limit on the port is set to 1 or if the limit (greater than 1) on the port has been reached.
- If a learning limit is already configured on a port when you enable the lockdown timeout feature, the configured limit will continue to apply. Existing blackholed entries are therefore not affected. If you enable this feature on a port with no configured learning limit, the default maximum learning limit (unlimited learning) is used.

Understanding the Lockdown Timer

The lockdown timer works in the following ways:

- When you enable this feature on a port, existing MAC entries for the port begin aging out based on the configured MAC lockdown timer value.
- If you move a device from one port to another, its MAC address entry is updated with the new port information, including the lockdown timer value configured for that port.
- If this feature is enabled on a port and you decrease the lockdown timer value for that port, all of the MAC *FDB* entries for that port will time out and be removed at the next polling interval.

- When you disable the lockdown timer on a port, existing MAC address entries for the port will time out based on the FDB aging period.
- The MAC lockdown timer takes effect when the port goes down/restarted.

Examples of Active and Inactive Devices

Figure 190 shows three devices (A, B, and C) connected through a hub to an Extreme Networks device with MAC lockdown timeout configured on the ports.

When each device starts sending traffic, the source MAC address of the device is learned and *FDB* entries are created. The MAC lockdown timer is set at 100 seconds.

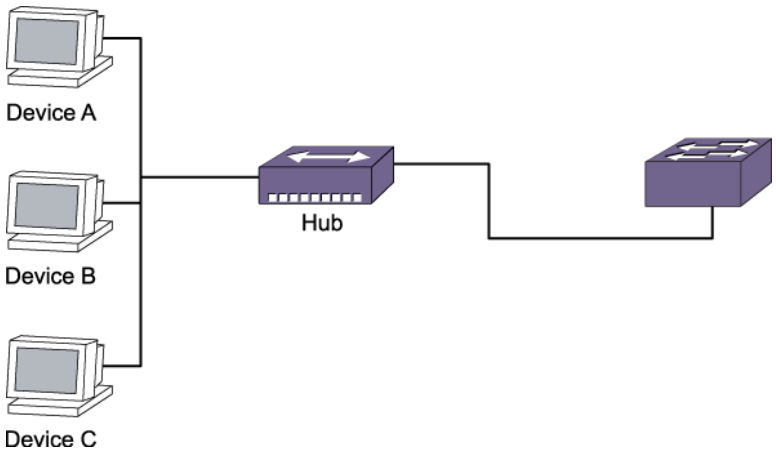


Figure 190: Devices Using MAC Address Lockdown

Device Inactivity for Less than the MAC Lockdown Timer

As long as a device continues to send traffic, the MAC entry for that device is refreshed, and the MAC lockdown timer corresponding to the MAC entry is refreshed.

Therefore, as long as the device is active, the timer does not expire. The traffic can be continuous or can occur in bursts within the MAC lockdown timeout duration for the port.

In this example, Device A starts sending traffic. When the MAC address of Device A is learned and added to the FDB, the MAC lockdown timer is started for this entry.

Device A stops sending traffic and resumes sending traffic after 50 seconds have elapsed. At this point the MAC entry for Device A is refreshed and the MAC lockdown timer is restarted.

Device Inactivity for Longer than the MAC Lockdown Timer

When a device stops sending traffic and does not resume within the MAC lockdown timer interval for the port, the MAC lockdown timer expires, and the MAC entry is removed from the FDB.

In this example, Devices A, B, and C start sending traffic. As each MAC address is learned, the MAC lockdown timer is started for each entry.

Device A stops sending traffic; Devices B and C continue sending traffic. After 100 seconds, the MAC lockdown timer for the Device A entry is removed from the FDB.

Because Devices B and C have continued to send traffic, their MAC entries continue to be refreshed and their MAC lockdown timers continue to be restarted.

Examples of Disconnecting and Reconnecting Devices

Figure 191 shows Device A connected to an Extreme Networks device with MAC lockdown timeout configured for the ports.

When Device A starts sending traffic, the source MAC address is learned on the port, the *FDB* entry is created, and the MAC lockdown timer is started for the entry. The MAC lockdown timer is set at 3,000 seconds.

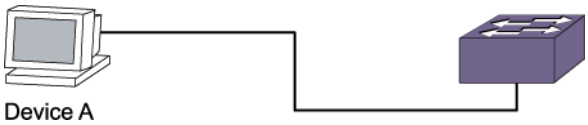


Figure 191: Single Device with MAC Lockdown Timeout

Disconnecting a Device

In this example, Device A is disconnected from the port, triggering a port-down action.

The MAC entry for Device A is removed from the hardware FDB; however, the MAC entry for the device is maintained in the software. The MAC lockdown timer for this entry starts when the port goes down.

After 3,000 seconds, the MAC entry for Device A is removed from the software.

Disconnecting and Reconnecting a Device

When Device A is disconnected from the port, the resulting port-down action causes the MAC entry for Device A to be removed from the hardware FDB.

The MAC entry in software is maintained, and the MAC lockdown timer is started for the port.

After only 1,000 seconds have elapsed, Device A is reconnected to the same port and starts sending traffic. A MAC entry is created in the hardware FDB, and the MAC lockdown timer is restarted for the MAC entry in the software.

If Device A is reconnected but does not send any traffic for 3,000 seconds, no MAC entry is created in the hardware FDB, and the MAC lockdown timer will expire after reaching 3,000 seconds.

Disconnecting and Reconnecting Devices with MAC Limit Learning

In this example, a MAC learning limit of 1 has also been configured on the ports in addition to the MAC lockdown timer of 3000 seconds.

When Device A is disconnected, the resulting port-down action removes the MAC entry for Device A from the hardware FDB. The MAC entry for Device A is maintained in the software, and the MAC lockdown timer for this entry is restarted when the port goes down.

After 1000 seconds, a different device is connected to the same port and starts sending traffic. Because the MAC learning limit is set to 1 and the MAC lockdown timer is still

running, the MAC address of the new device is not learned. Instead, the new MAC address is blackholed in the hardware.

When the MAC lockdown timer for Device A expires, its MAC entry is removed from the software. If the new device is still connected to the same port and sends traffic, the MAC address for the new device is learned and added to the FDB. The MAC lockdown timer for the new device is started, and the blackhole entry that was created for this device is deleted.

Example of Port Movement

Figure 192 shows Device A connected to port X.

Port X has a MAC lockdown timer setting of 100 seconds, and port Y has a MAC lockdown timer setting of 200 seconds.

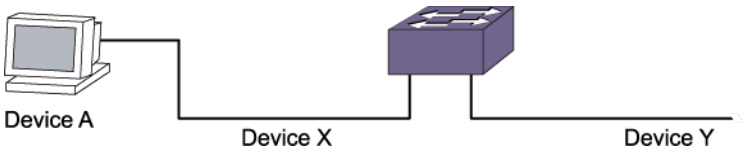


Figure 192: Port Movement with MAC Lockdown Timeout

Device A starts sending traffic on port X. The MAC address for Device A is learned and added to the *FDB*, and the MAC lockdown timer (100 seconds) is started for this entry.

After 50 seconds, Device A is disconnected from port X and connected to port Y where it begins sending traffic. When Device A starts sending traffic on port Y, the existing MAC entry for Device A is refreshed, and port X in the entry is replaced with port Y. At the same time, the MAC lockdown timer for the entry is restarted for a duration of 200 seconds (the configured MAC lockdown timer setting for port Y).

Configuring MAC Address Lockdown with Timeout

- Configure the MAC lockdown timeout value on one or more specified ports, or on all ports using the following command:

  ```
  configure mac-lockdown-timeout ports [all | port_list] aging-time seconds
  ```

Enabling and Disabling MAC Address Lockdown with Timeout

- Enable the MAC lockdown timeout feature on one or more specified ports, or on all ports. using the command:

  ```
  enable mac-lockdown-timeout ports [all | port_list]
  ```

- Disable the MAC lockdown timeout feature on one or more specified ports, or on all ports using the command:

  ```
  disable mac-lockdown-timeout ports [all | port_list]
  ```

Displaying MAC Address Lockdown Information

- Display configuration information about the MAC lockdown timeout feature using the command:

  ```
  show mac-lockdown-timeout ports [all | port_list]
  ```

Output from this command includes the configured timeout value and whether the feature is enabled or disabled.

- Display the MAC entries learned on one or more ports, or on all ports using the command:

  ```
  show mac-lockdown-timeout fdb ports [all | port_list]
  ```

  Output from this command also lists the aging time of the port.

MAC Security with Pre-shared Key Authentication

Media Access Control Security (MACsec) as defined by *IEEE Std 802.1X-2010 Port-Based Network Access Control* (control plane) and *IEEE Std 802.1AE-2006 Media Access Control (MAC) Security* (data plane) allows authorized systems that attach to, and interconnect, LANs in a network to maintain confidentiality of transmitted data and to take measures against frames transmitted or modified by unauthorized devices.

Overview

MACsec is configured on a per-port basis to protect point-to-point links between switches. Mutual authentication is achieved by provisioning the same set of credentials (pre-shared key) on each end of a link.

Prior to authentication, all port traffic is blocked. After authentication, all port traffic is protected by the GCM-AES-128 cipher suite by default, or optionally, by GCM-AES-256. MACsec operates at Layer 2 and is therefore protocol agnostic, encrypting everything it passes. Because encryption takes place at the hardware level, line-rate traffic passes

with low latency, but due to additional MACsec headers, some throughput drop occurs. MACsec operates on a hop-by-hop basis, allowing for deep packet inspection.

Note
The following table lists the switches/ports that support the optional GCM-AES-256 cipher.

Table 138: Switches/Ports that Support the GCM-AES-256 Cipher

Platform	Ports	LRM/ MACsec Adapter Required?
ExtremeSwitching 5320-24T-24S-4XE-XT	All ports except stacking ports.	No
ExtremeSwitching 5320	All ports of all models except stacking ports.	No
ExtremeSwitching 5420	All ports of all models except stacking ports.	No
ExtremeSwitching 5520	All ports, except 5520-VIM-4X and 24X 10G ports	No
ExtremeSwitching 5720	All ports of all models except stacking ports.	No
Extreme 7520-48YE-8CE	All front panel ports.	No

Authentication is provided by pre-shared-keys (PSK), which consist of a public secure connectivity association key name (CKN) and a private secure connectivity association key (CAK). Each PSK is configured against a connectivity-association namespace. Each connectivity-association can be applied to one or more MACsec-capable ports. Each port may belong to only one connectivity-association.

Note
When MACsec is enabled, every protected packet is prefixed with an 8-byte (include-sci disable) or 16-byte (include-sci enable) SecTAG and suffixed with a 16-byte Integrity Check Value (ICV). If the average packet size on a port is small, then these 24 to 32 extra bytes per packet have a non-trivial impact on throughput. This is a function of the protocol, and is not a factor of this implementation.

Note
MACsec-enabled port mirroring for egress traffic is not supported on 5420 switches.

MAC Security with Pre-shared Keys Authentication Limitations

The following limitations apply to the MAC Security (MACsec) with Pre-shared Keys (PSK) Authentication:

- This initial release of MACsec only implements section point-to-point LANs within a secured network as described in *Clause 7.4 MACsec to support Infrastructure LANs* of the much broader standard outlined in *IEEE802.1X-2010 Port-Based Access Control*. All other sections and clauses are not supported.

  Note
  MACSec between customer edges over L2VPN is supported on untagged access ports.

- MACsec is only configurable using CLI commands. There is no SNMP access to the two MACsec MIBs defined by IEEE: IEEE8021X-PAE-MIB and IEEE8021-SECY-MIB.
- MACsec is not supported on ports with stacking enabled.
- MACsec is not supported on Extended Edge Switching ports.
- Hot swapping LRM/MACsec Adapters is not supported. MACsec must be disabled before hot swapping.
- The LRM/MACsec Adapters cannot be connected across slots on a stack.
- The 5320 and 5720 switches use their native MACsec. MACsec is not supported on an LRM/MACsec Adapter when connected to this switch.
- On the 5720 switch, corresponding 5720-VIM-6YE port groups need to be configured to 3x10G partition to support an LRM/MACsec Adapter and 10G-LRM optics.
- MACSec is not supported on MLAG ports in 5420 and 5320 stacked switches.

MAC Security with Pre-shared Keys Authentication Supported Platforms

MAC Security (MACsec) with Pre-shared Keys (PSK) authentication is supported built-in the following platforms.

Platform	Ports
5120	All ports of all models except stacking ports.
5320	All ports of all models, except 5320-24T-4X-XT, and except stacking ports.
5420	All ports of all models except stacking ports.
5520	All ports, except 5520-VIM-4X and 5520-24X 10G ports
5720	All ports of all models except stacking ports.
7520-48YE-8CE	All front-panel ports.

Note
The MACsec feature requires the installation of the MAC Security feature pack license.

Note
When an LRM/MACsec adapter is connected to 5320 or 5720 switch, MACsec will be provided by the native switch port rather than by the adapter.

Note
On 5420 and 5320 switches, when OnePolicy is enabled, the MACsec link may fail due to the unavailability of ACL resources. The only way to overcome this is to reduce the number of ACLs reserved by Policy by using the `configure policy resource-profile` **`default profile-modifier`** command.

Note
Remote mirroring (ERSPAN) will not work as expected on 5420 and 5320 switches when the remote mirroring end-point is reachable using a MACSEC uplink. This is due to HW limitations on these platforms.

Configuring MAC Security with Pre-shared Keys Authentication

To configure MAC Security (MACsec) with pre-shared keys (PSKs) authentication on a switch port:

1. Install a MACsec license key (if one is not already installed) by using the following command:

   `install license` **`file`** *`filename`* {**`slot`** *`slot`*}

   Note
   ExtremeSwitching Universal switches natively support MACsec and do not require an adapter.

2. Create a connectivity-association (CA) object that holds MACSec authentication data (secure connection association key (CAK) and secure connection association key name (CKN) pair, which makes up the PSK on each port enabled for MKA by using the following command:

   `create macsec` **`connectivity-association`** *`ca_name`* **`pre-shared-key ckn`** *`ckn`* **`cak`** [**`encrypted`** *`encrypted_cak`* | *`cak`*]

3. Optionally, modify MACsec replay protection packet window, which allows for dropping of out-of-order packets received on a port by using the following command:

   `configure macsec` **`replay-protect`** [*`window_size_in_packets`* | **`disable`**] **`ports`** *`port_list`*

   The replay protection feature provides for the dropping of out-of-order packets received on a port. The window size is set to 0 by default, meaning any packet received out-of-order is dropped. Setting the window size to non-zero sets the range of sequence numbers that are tolerated, to allow receipt of packets that have been misordered by the network. If replay protection is disabled, packet sequence numbers are not checked and out-of-order packets are not dropped.

4. Optionally, configure a port's priority for becoming a key server by using the following command:

```
configure macsec mka actor-priority actor_priority ports port_list
```

5. Optionally, configure the include-SCI flag (to ensure interoperability with third-party devices that do not decode encrypted MACsec packets when the SCI is not present) using the following command:

   ```
   configure macsec include-sci [enable | disable] ports port_list
   ```

6. Optionally, change the MACsec cipher suite by using the following command:

   ```
   configure macsec cipher-suite [gcm-aes-128 | gcm-aes-256] ports
   port_list
   ```

   Note
   Universal Switches support both GCM-AES-128 and GCM-AES-256 cipher natively on all ports except for Universal ports U1 and U2.

7. Enable MACsec authentication on the desired ports by using the following command:

   ```
   configure macsec connectivity-association ca_name [pre-shared-key {ckn
   ckn} {cak [encrypted encrypted_cak] | cak} | ports [port_list] [enable
   | disable]]
   ```

   Use the *ca_name* set up in Step 2 on page 1559, use the enable option, and designate the port(s).

Important
After enabling MACsec, if you change the actor priority, replay protection window, mka life-time, or include-SCI flag, you must run the `configure macsec initialize ports port_list` afterward. Otherwise, the change is not accepted.

To **delete** a previously created CA object, use the following command:

```
delete macsec connectivity-association ca_name
```

To **clear MACsec counters**, use the following command:

```
clear macsec counters {ports [port_list]}
```

To **reset** the MACsec Key Agreement protocol state machine on one or more ports, use the following command:

```
configure macsec initialize ports port_list
```

Issuing this command resets the MKA state machine, which in turn deletes any secured channels and their secure association keys (SAKs). This command is also used to apply MACsec configuration changes (mka actor-priority, include-sci, replay-protect, mka life-time) to an already enabled port. All traffic is blocked until MKA renegotiates a new set of keys and those keys are installed. For more information, see *IEEE802.1X-2010 Clause 12.9.3 Initialization*.

Displaying MACsec Information

To display a system-wide view of MACsec, use the following command:

show macsec

To display a global summary of MACsec capabilities and status for all or a specified CA, use the following command:

show macsec { **connectivity-association** {*ca_name*}

To display per-port MKA and MACsec data in tabular format, use the following command:

show macsec **ports** *port-list* **usage**

To display a table of all configurable parameters, use the following command:

show macsec **ports** *port-list* **configuration**

To display configuration, status, and statistics for both MKA and MACsec, use the following command:

show macsec **ports** *port-list* **detail**

To display the number of ports that have MACsec enabled and the maximum number of ports allowed per slot, use the following command:

show macsec **usage**

To display the transmitted and dropped packets for each MACsec engine, use the following command:

show ports **macsec-engines** [**qosmonitor** | **congestion**] {**no-refresh** | **refresh**}

Displaying LRM/MACsec Adapter Information

To display that a LRM/MACsec adapter is connected to a port, use either of the following commands:

show ports {**mgmt** | *port_list* | **tag** *tag*} **configuration** {**no-refresh** | **refresh**}

show port {**mgmt** |*port_list* | **tag** *tag*} **information** {**detail**} using the detail option.

Troubleshooting MAC Security

If you find that a connection that you have set up MAC Security (MACsec) on is not secure, use the following information to troubleshoot the issue.

1. Verify that the port is enabled.

   show ports {*port_list* | **tag** *tag*} {**no-refresh** | **refresh**}

   E = Enabled.

2. Verify that the link is up before MACsec is enabled.
   `show ports {port_list | tag tag} {no-refresh | refresh}`

   A = Active.
3. Verify MACsec license is installed.
   `show licenses {[slot slot |all]} {detail}`

```
# show license
Enabled License Level:
        Advanced Edge
Enabled Feature Packs:
        MACsec
```

4. Verify that MACsec is enabled.
   `show macsec ports port-list usage`
5. Verify that pre-shared-keys (PSKs) are identical by looking for the event: `PortCKNMisMatch`.

   To have PortCKNMisMatch logged by default, change the severity of logged MACsec events from "error" to "notice":

```
configure log filter DefaultFilter add events macsec severity notice
```

   Look for the following event.

```
<Noti:MACsec.MKA.PortCKNMisMatch> On port 50, Secure Connectivity Association Key Name
(CKN) is not included in local Secure Connectivity Association (CA). A possible CKN
mismatch.
```

   `show log {messages [memory-buffer | nvram]} {events {event-condition | event-component]} {severity severity {only}} {starting [date date time time | date date | time time]} {ending [date date time time | date date | time time]} {match regex} {chronological}`
6. Verify MACsec Key Agreement PDUs (MKPDUs) are being transmitted and received.
   `show macsec ports port-list usage`

   Verify that local and peer message numbers are incrementing.

Note

If a warning message in the log indicates that LRM/MACsec adapter port firmware is out-of-date:

```
<Warn:HAL.Port.Warning> LRM/MACsec Adapter port firmware on port 2:16 is out of
date, do 'install firmware lrm-macsec-adapter' to update.
```

use the following command to update the firmware:

`install firmware {{force} {slot slot_number} | lrm-macsec-adapter ports [port_list | all]}`

Note

If MACsec link flap periodically occurs, try to increase "mka life-time". This solution works best with lower-end switches.

Alternate MAC Address Support

Version 33.2.1 adds support for configuring an alternate MAC address using the command line. This command enables you to set a new MAC address that differs

from the original one stored in the Electrically Erasable Programmable Read-Only Memory (EEPROM). The alternate MAC address is saved in a persistent "well-known info" database, which is used by the operating system for switch management tasks. When restarting, the operating system detects if an alternate MAC address has been configured. If it has, the operating system overwrites the factory MAC address.

This feature is only applicable to standalone switches and does not support switch stacks.

Supported Platforms

All platforms.

Limitations

The following limitations apply to this feature:

- You cannot configure a Locally Administered Address (LAA) as the alternate MAC address.
- You cannot configure multicast MAC addresses as the alternate MAC address.

New CLI Command

The following new CLI command is introduced in this version:

```
configure switch mac-address
```

DHCP Server

The *DHCP (Dynamic Host Configuration Protocol)* server implementation in ExtremeXOS only supports address allocation on the primary IP interface of the configured *VLAN*. That is, all DHCP clients residing on a bridging domain have IP addresses belonging to the primary subnet. To add a host on secondary subnet, you must manually configure the IP address information on that host.

Enabling and Disabling DHCP

DHCP is enabled on a per port, per *VLAN* basis.

- Enable or disable DHCP on a port in a VLAN using the commands:

  ```
  enable dhcp ports port_list vlan vlan_name
  disable dhcp ports port_list vlan vlan_name
  ```

Configuring the DHCP Server

The following commands allow you to configure the *DHCP* server included in the switch.

Note
Configuring both the DHCP server and the DHCP client on the same VLAN and the same ports is not recommended.

The parameters available to configure include the IP address range, IP address lease, and multiple DHCP options. Until ExtremeXOS 15.1, the DHCP server had a limited set

of known DHCP options it could send out on request, i.e., Default Gateway, DNS, and WINS server(s). General option support has been added in ExtremeXOS 15.2. This allows you to add support for any option needed, with no ExtremeXOS code changes. The three options mentioned above can also be overwritten to support a larger number of servers, if needed. This feature allows the switch administrator to add an option based on DHCP option code value, and support various was of setting the value.

1. Configure the range of IP addresses assigned by the DHCP server using the command:
   `configure vlan vlan_name dhcp-address-range ipaddress1 - ipaddress2`
2. Remove the address range information using the command:
   `unconfigure vlan vlan_name dhcp-address-range`
3. Set how long the IP address lease assigned by the server exists using the command:
   `configure vlan vlan_name dhcp-lease-timer lease-timer`

   Note
   The ExtremeXOS DHCP server allows the configuration of a DHCP lease timer value greater than two seconds only. The timer value range is 3–4294967295. If the DHCP lease timer is not configured, the ExtremeXOS DHCP server offers an IP address with the default lease time of 7200 seconds.

   Note
   Maximum number of DHCP leases supported by the ExtremeXOS DHCP server is limited by the number of VLANs and IP address range configured on that VLAN, and there is no hardware/software limit for the DHCP leases.

   Note
   The maximum number of DHCP servers supported by ExtremeXOS is limited by the number of VLANs the ExtremeXOS switch supports.

4. To set the generic DHCP option code, default gateway, Domain Name Servers (DNS) addresses, or Windows Internet Naming Service (WINS) server, use the following command:
   `configure {vlan} vlan_name dhcp-options [code option_number [16-bit value1 {value2 {value3 {value4}}} | 32-bit value1 {value2 {value3 {value4}}} | flag [on | off] | hex string_value | ipaddress ipaddress1 {ipaddress2 {ipaddress3 {ipaddress4}}} | string string_value] | default-gateway | dns-server {primary | secondary} | wins-server] ipaddress`
5. To remove the generic DHCP option code, default gateway, DNS server addresses, and WINS server information for a particular *VLAN*, use the following command:
   `unconfigure {vlan} vlan_name dhcp-options {[ default-gateway | dns-server {primary | secondary} | wins-server]}`
6. Remove all the DHCP information for a particular VLAN using the command:
   `unconfigure vlan vlan_name dhcp`

   You can clear the DHCP address allocation table selected entries, or all entries.

7. Clear entries using the command:
   `clear vlan vlan_name dhcp-address-allocation [[all {offered | assigned | declined | expired}] | ipaddress]`

   You would use this command to troubleshoot IP address allocation on the VLAN.

Displaying DHCP Information

- Display the *DHCP* configuration, including the DHCP range, DHCP lease timer, network login lease timer, DHCP-enabled ports, IP address, MAC address, and time assigned to each end device using the command:
  `show dhcp-server {vlan vlan_name}`
- View only the address allocation of the DHCP server on a *VLAN* using the command:
  `show {vlan} vlan_name dhcp-address-allocation`
- View only the configuration of the DHCP server on a VLAN using the command:
  `show {vlan} vlan_name dhcp-config`

IP Security

This section describes a collection of IP security features implemented in ExtremeXOS software.

If you configure any of the features described in this section, you can enhance your network security by controlling which hosts are granted or not granted access to your network.

Figure 193 displays the dependencies of IP security. Any feature that appears directly above another feature depends on it. For example, to configure ARP validation, you must configure *DHCP* snooping and trusted DHCP server.

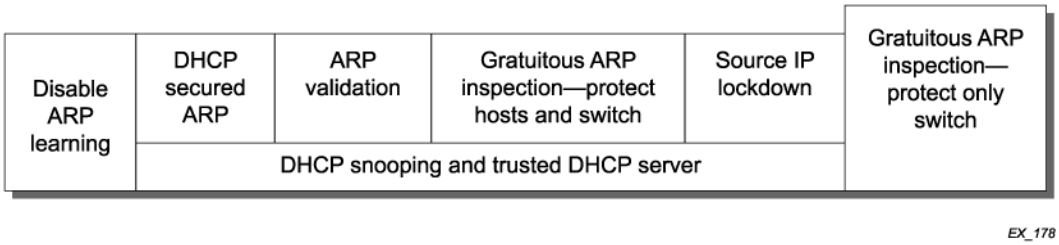


Figure 193: IP Security Dependencies

Note
IP security features are supported on link aggregation ports with the exception of DHCP snooping with the **block-mac** option and source IP lockdown. You can enable IP security on pre-existing trunks, but you cannot make IP security-enabled ports into trunks without first disabling IP security.

DHCP Snooping and Trusted DHCP Server

A fundamental requirement for most of the IP security features described in this section is to configure *DHCP* snooping and trusted DHCP server.

DHCP snooping enhances security by filtering untrusted DHCP messages and by building and maintaining a DHCP bindings database. Trusted DHCP server also

enhances security by forwarding DHCP packets from only configured trusted servers within your network.

The DHCP bindings database contains the IP address, MAC Address, *VLAN* ID, and port number of the untrusted interface or client. If the switch receives a DHCP ACK message and the IP address does not exist in the DHCP bindings database, the switch creates an entry in the DHCP bindings database. If the switch receives a DHCP RELEASE, NAK or DECLINE message and the IP address exists in the DHCP bindings database, the switch removes the entry.

You can enable DHCP snooping on a per port, per VLAN basis and trusted DHCP server on a per-vlan basis. If configured for DHCP snooping, the switch snoops DHCP packets on the indicated ports and builds a DHCP bindings database of IP address and MAC address bindings from the received packets. If configured for trusted DHCP server, the switch forwards only DHCP packets from the trusted servers. The switch drops DHCP packets from other DHCP snooping-enabled ports.

In addition, to prevent rogue DHCP servers from farming out IP addresses, you can optionally configure a specific port or set of ports as trusted ports. Trusted ports do not block traffic; rather, the switch forwards any DHCP server packets that appear on trusted ports. When configured to do so, the switch drops packets from DHCP snooping-enabled ports and causes one of the following user-configurable actions: disables the port temporarily, disables the port permanently, blocks the violating MAC address temporarily, blocks the violating MAC address permanently, and so on.

Note

When IP security DHCP snooping is enabled on a VLAN, if ports are removed from the VLAN, the IP security snooping configuration is removed. If ports are added back to the VLAN, you must manually enable the IP security snooping configuration again. This is true for both plain and LAG ports.

Also, if LAG is unconfigured, the IP security configuration is removed for those ports.

Also, if a new port is added to a VLAN that has IP security snooping enabled on it already for other ports, you must manually enable IP security snooping again for the new added port. IP security commands are not automatically applied to added ports to a VLAN. They must be manually configured again.

Note

Beginning with version 32.5, the functionality of the option *port all* in the commands **enable ip-security dhcp-snooping dynamic vlan ports all violation-action drop-packet** or **enable ip-security dhcp-snooping vlan v2 ports all violation-action drop-packet** is enhanced to cover all the _future_ member ports of a VLAN or _future_ VLANs.

If the port is dynamically removed or added by any module, for example Netlogin or ONEPolicy, the *ports all* option enables IP-security snooping on the ports corresponding to the VLAN it adds.

Configuring DHCP Snooping

DHCP snooping is disabled on the switch by default.

- To enable DHCP snooping on the switch, use the command:

```
enable ip-security dhcp-snooping [dynamic | {vlan} vlan_name] ports
[all | ports] violation-action [drop-packet {[block-mac | block-port]
[duration duration_in_seconds | permanently] | none}] {snmp-trap}
```

Note
Snooping IP fragmented DHCP packets is not supported.

The violation action setting determines what action(s) the switch takes when a rogue DHCP server packet is seen on an untrusted port or the IP address of the originating server is not among those of the configured trusted DHCP servers.

The DHCP server packets are DHCP OFFER, ACK and NAK. The following list describes the violation actions:

block-mac

The switch automatically generates an *ACL* to block the MAC address on that port. The switch does not blackhole that MAC address in the *FDB*. The switch can either temporarily or permanently block the MAC address.

block-port

The switch blocks all traffic on that port by disabling the port either temporarily or permanently.

none

The switch takes no action to drop the rogue DHCP packet or block the port, and so on. In this case, DHCP snooping continues to build and manage the DHCP bindings database and DHCP forwarding will continue in hardware as before. This option can be used when the intent is only to monitor the IP addresses being assigned by the DHCP server.

Note
You must enable DHCP snooping on both the DHCP server port as well as on the client port. The latter ensures that DHCP client packets (DHCP Request, DHCP Release etc.) are processed appropriately.

Note
DHCP snooping does not work when the client and server are in different VRs and server reachability is established by inter-VR leaked routes on client VR.

Note
Enabling DHCP snooping and source IP lockdown on the same port applies ACL rules with the same match conditions, but different actions. The rule with deny action takes precedence, so packets are dropped if the these ACL rules are installed on different slices. Many factors influence which slice rules are installed on. To see which slice these rules are installed on, use the command `show access-list usage acl-slice port port` or `show access-list usage acl-rule port port` .

Any violation that occurs causes the switch to generate an EMS log message. You can configure to suppress the log messages by configuring EMS log filters. For more information about EMS, see Using the Event Management System/Logging on page 1750.

- To disable DHCP snooping on the switch, use the command:
  `disable ip-security dhcp-snooping [dynamic | {vlan} vlan_name] ports [all | ports]`

Configuring Trusted DHCP Server

- Configure a trusted *DHCP* server on the switch using the command:
  `configure trusted-servers [dynamic vlan_id |{vlan} vlan_name] add server ip_address trust-for dhcp-server`

  If you configure one or more trusted ports, the switch assumes that all DHCP server packets on the trusted port are valid. You can configure a maximum of eight trusted DHCP servers per VLAN on the switch.

  For more information about configuring trusted ports, see Configuring Trusted DHCP Ports on page 1568.
- Delete a trusted DHCP server using the command:
  `configure trusted-servers [dynamic vlan_id |vlan vlan_name] delete server ip_address trust-for dhcp-server`

Configuring Trusted DHCP Ports

Trusted ports do not block traffic; rather, the switch forwards any *DHCP* server packets that appear on trusted ports. Depending on your DHCP snooping configuration, the switch drops packets and can disable the port temporarily, disable the port permanently, block the MAC address temporarily, block the MAC address permanently, and so on.

- Enable trusted ports on the switch using the command:
  `configure trusted-ports [ports|all] trust-for dhcp-server`
- Disable trusted ports on the switch using the command:
  `unconfigure trusted-ports [ports |all] trust-for dhcp-server`

Displaying DHCP Snooping and Trusted Server Information

- Display the *DHCP* snooping configuration settings using the command:
  `show ip-security dhcp-snooping {vlan} vlan_name`

Below is sample output from this command:

```
DHCP Snooping enabled on ports: 1:2, 1:3, 1:4, 1:7, 1:9
Trusted Ports: 1:7
Trusted DHCP Servers: None
-------------------------------------------
Port            Violation-action
-------------------------------------------
1:2             none
1:3             drop-packet
1:4             drop-packet, block-mac permanently
1:7             none
1:9             drop-packet, snmp-trap
```

- Display the DHCP bindings database using the command:

  `show ip-security dhcp-snooping entries {vlan} vlan_name`

  Below is sample output from this command:

```
-------------------------------------------
Vlan: dhcpVlan
-------------------------------------------
Server    Client
IP Addr          MAC Addr            Port      Port
-------          --------            ------    ------
172.16.100.9     00:90:27:c6:b7:65   1:1       1:2
```

Clearing DHCP Snooping Entries

- To clear Existing *DHCP* snooping entries, use the command:

  `clear ip-security dhcp-snooping entries {vlan} vlan_name`

  Note
  This will also clear out any associated source IP lockdown and DHCP secured ARP entries.

Configuring the DHCP Relay Agent Option (Option 82) at Layer 2

This section describes how to configure the *DHCP* Relay agent option for Layer 2 forwarded DHCP packets.

The DHCP relay agent option feature inserts a piece of information, called option 82, into any DHCP request packet that is to be relayed by the switch. Similarly, if a DHCP reply received by the switch contains a valid relay agent option, the option will be stripped from the packet before it is relayed to the client. This is a Layer 2 option that functions only when the switch is not configured as a Layer 3 BOOTP relay.

The **Agent remote ID sub-option** always contains the Ethernet MAC address of the relaying switch. You can display the Ethernet MAC address of the switch by issuing the show switch command.

The contents of the inserted option 82 sub-options is as follows:

Table 139: Contents of the Inserted Option 82 Sub-options

Code (1 byte)	Length (1 byte)	Sub-Option (1 byte)	Length (1 byte)	Value (1-32 bytes)	Sub-Option (1 byte)	Length (1 byte)	Switch MAC address (6 bytes)
82		1 (Circuit ID)	1-32	vlan_info-port_info	2 (Remote ID)	6	

To enable the DHCP relay agent option at Layer 2, use the following command:

```
configure ip-security dhcp-snooping information option
```

Note
When DHCP relay is configured in a DHCP snooping environment, the relay agent IP address should be configured as the trusted server.

When DHCP option 82 is enabled, two types of packets need to be handled:

- DHCP Request: When the switch (relay agent) receives a DHCP request, option 82 is added at the end of the packet. If the option has already been enabled, then the action taken depends on the configured policy (drop packet, keep existing option 82 value, or replace the existing option). Unless configured otherwise using the `configure ip-security dhcp-snooping information circuit-id vlan-information` *`vlan_info`* `[`**`dynamic`**` | {`**`vlan`**`}` *`vlan_name`* ` | `**`all`**`]` , the vlan_info portion of the circuit ID added will be the *VLAN* ID of the ingress VLAN.
- DHCP Reply: When the option 82 information check is enabled, the packets received from the DHCP server are checked for option 82 information. If the remote ID sub-option is the switch's MAC address, the packet is sent to the client; if not, the packet is dropped. If the check is not enabled. The packets are forwarded as-is.

To disable the DHCP relay agent option, use the following command:

```
unconfigure ip-security dhcp-snooping information option
```

In some instances, a DHCP server may not properly handle a DHCP request packet containing a relay agent option.

To prevent DHCP reply packets with invalid or missing relay agent options from being forwarded to the client, use the following command:

```
configure ip-security dhcp-snooping information check
```

To disable checking of DHCP replies, use this command:

```
unconfigure ip-security dhcp-snooping information check
```

A DHCP relay agent may receive a client DHCP packet that has been forwarded from another relay agent.

If this relayed packet already contains a relay agent option, then the switch will handle this packet according to the configured DHCP relay agent option policy. The possible

actions are to replace the option information, to keep the information, or to drop packets containing option 82 information. To configure this policy, use the following command:

```
configure ip-security dhcp-snooping information policy [drop | keep | replace]
```

The default relay policy is replace.

To configure the policy to the default, use this command:

```
unconfigure ip-security dhcp-snooping information policy
```

The Layer 2 relay agent option allows you to configure the circuit ID on a VLAN or port basis., the Circuit-ID can contain a variable length (up to 32 bytes long) ASCII string with the following format:

```
<VLAN Info>-<Port Info>
```

If the configuration of either VLAN Info or Port Info causes the total string length of <VLAN Info>-<Port Info> to exceed 32 bytes, then it is truncated to 32 bytes. The string is not NULL terminated, since the total circuit ID length is being specified.

For a DHCP client packet ingressing on a VLAN with the VLAN ID equal to 200 and the ingress port at 3:5, the following are true:

- When neither VLAN Info or Port Info is specified, circuit ID value is = 200-3005
- When VLAN Info is configured to SomeInfo and Port Info is not specified, the circuit ID value is SomeInfo-3005
- When VLAN Info is not specified and Port Info is configured to User1, the circuit ID value is 200-User1
- When VLAN Info is configured to SomeInfo and Port Info to User1, the circuit ID value is SomeInfo-User1

VLAN Info is configurable per VLAN.

When not explicitly configured for a VLAN, VLAN Info defaults to the ASCII string representation of the ingress VLAN ID. To configure the circuit ID on a VLAN, use the following command:

```
configure ip-security dhcp-snooping information circuit-id vlan-information vlan_info [dynamic | {vlan} vlan_name | all]
```

To unconfigure the circuit ID on a VLAN, use the following command:

```
unconfigure ip-security dhcp-snooping information circuit-id vlan-information [dynamic | {vlan} vlan_name |all]
```

Port Info is configurable.

When not explicitly configured for a port, port info defaults to the ASCII representation of the ingress port's *SNMP* ifIndex. To configure the port information portion of the circuit-ID, use the following command:

```
configure ip-security dhcp-snooping information circuit-id port-
information port_info port port
```

To unconfigure the port information portion of the circuit-ID, use the following command:

```
unconfigure ip-security dhcp-snooping information circuit-id port-
information ports [port_list | all]
```

Note

When this feature is enabled, all DHCP traffic must be forwarded in slowpath only, which means that this feature functions only in the context of IP Security and only on interfaces where DHCP snooping is enabled in enforcement (violation-action of 'drop') mode. In other words, with DHCP snooping not configured with a violation-action of 'none' (which is pure monitoring mode).

Configuring DHCP Binding

The *DHCP* bindings database contains the IP address, MAC address, *VLAN* ID, and port number of the client. You can add or delete the static IP to the MAC DHCP binding entries using the following commands:

configure ip-security dhcp-bindings add

configure ip-security dhcp-bindings delete

You can specify the storage details of the DHCP binding database. Use the following commands to specify the DHCP binding database location, filename, write-interval, and write threshold limits:

```
configure ip-security dhcp-bindings storage filename
```

configure ip-security dhcp-bindings storage location

configure ip-security dhcp-bindings storage

You can upload the DHCP binding database periodically by enabling the DHCP binding restoration. Binding write intervals occur in minutes, 5 to 1440 (24 hours). The default is 30 minutes.

Upload the latest DHCP binding database using the upload command:

enable ip-security dhcp-bindings restoration

You can also upload the DHCP binding database by the number of DHCP entries (the write-threshold is 25 to 200).

The periodic backup of the DHCP binding database can be disabled using the following command:

disable ip-security dhcp-bindings restoration

Note
There is no command to unconfigure the DHCP binding storage server details. To disable the DHCP binding storage server details, use the preceding command.

For information about configuring option 82 at Layer 3, see Configuring the DHCP Relay Agent Option (Option 82) at Layer 3 on page 923.

Note
When configuring static DHCP binding entries, DHCP binding restoration needs to be configured.

Example of Option 82 Configuration

The following example describes Option 82 configuration for circuit ID fields.

```
create vlan v1
conf v1 add ports 21
enable ip-security dhcp-snooping v1 ports all violation-action drop-packet
configure trusted-ports 21 trust-for dhcp-server
conf ip-security dhcp-snooping information option
conf ip-security dhcp-snooping information check
conf ip-security dhcp-snooping information circuit-id vlan-information ServiceProvider-1
v1
conf ip-security dhcp-snooping information circuit-id port-information cutomer-1 port 1
conf ip-security dhcp-snooping information circuit-id port-information cutomer-2 port 2
CLI display output
==================

* switch # sh ip-security dhcp-snooping v1
DHCP Snooping enabled on ports: 21
Trusted Ports: 21
Trusted DHCP Servers: None
Bindings Restoration     : Disabled
Bindings Filename        :
Bindings File Location   :
Primary Server  : None
Secondary Server: None
Bindings Write Interval  : 30 minutes
Bindings last uploaded at:
-----------------------------------
Port            Violation-action
-----------------------------------
21              drop-packet

* switch # show ip-security dhcp-snooping information-option
Information option insertion: Enabled
Information option checking : Enabled
Information option policy   : Replace
* switch #

* switch # sh ip-security dhcp-snooping information-option circuit-id vlan-information
Vlan            Circuit-ID vlan information string
----            ----------------------------------
Default         1 (Default i.e. vlan-id)
Mgmt            4095 (Default i.e. vlan-id)
v1              ServiceProvider-1
```

```
Note: The full Circuit ID string has the form '<Vlan Info>-<Port Info>'
* switch


* switch # sh ip-security dhcp-snooping information-option circuit-id port-information
ports all
Port            Circuit-ID Port information string
----            ----------------------------------
1               cutomer-1
2               cutomer-2
3               1003
4               1004
5               1005
6               1006
7               1007
8               1008
9               1009
10              1010
11              1011
12              1012
13              1013
14              1014
15              1015
16              1016
17              1017
18              1018
19              1019
20              1020
21              1021
22              1022
23              1023
24              1024
25              1025
26              1026
Note: The full Circuit ID string has the form '<Vlan Info>-<Port Info>'
* switch #
```

Source IP Lockdown

Another type of IP security prevents IP address spoofing by automatically placing source IP address filters on specified ports. This feature, called *source IP lockdown*, allows only traffic from a valid *DHCP*-assigned address obtained by a DHCP snooping-enabled port to enter the network. In this way, the network is protected from attacks that use random source addresses for their traffic. With source IP lockdown enabled, end systems that have a DHCP address assigned by a trusted DHCP server can access the network, but traffic from others, including those with static IP addresses, is dropped at the switch.

Source IP lockdown is linked to the “DHCP snooping” feature. The same DHCP bindings database created when you enable DHCP snooping is also used by source IP lockdown to create ACLs that permit traffic from DHCP clients. All other traffic is dropped. In addition, the DHCP snooping violation action setting determines what action(s) the switch takes when a rogue DHCP server packet is seen on an untrusted port.

When source IP lockdown is enabled on a port, a default *ACL* is created to deny all IP traffic on that port. Then an ACL is created to permit DHCP traffic on specified ports. Each time source IP lockdown is enabled on another port, the switch creates ACLs to allow DHCP packets and to deny all IP traffic for that particular port.

Source IP lockdown is enabled on a per-port basis; it is not available at the *VLAN* level. If source IP lockdown is enabled on a port, the feature is active on the port for all VLANs to which the port belongs.

Note
The source IP lockdown feature works only when hosts are assigned IP address using DHCP; source IP lockdown does not function for statically configured IP addresses.

Note
Source IP lockdown cannot be enabled on Load sharing ports.

The source IP lockdown ACLs listed in table are applied per port (in order of precedence from highest to lowest).

Table 140: Source IP Lockdowns Applied Per-port

ACL Name	Match Condition	Action	When Applied	Comments
esSrcIpLockdown_<portIfIndex>_<source IP in hex>	Source IP	Permit	Runtime	Multiple ACLs of this type can be applied, one for each permitted client.
esSrcIpLockdown_<portIfIndex>_1	Proto UDP, Dest Port 67	Permit	Configuration time	
esSrcIpLockdown_<portIfIndex>_2	Proto UDP, Dest Port 68	Permit	Configuration time	
esSrcIpLockdown_<portIfIndex>_3	Ethertype ARP	Permit	Configuration time	
esSrcIpLockdown_<portIfIndex>_4	All	Deny + count	Configuration time	

The counter has the same name as that of the rule of the catch-all ACL, so the counter is also named esSrcIpLockdown_<portIfIndex>_4.

Configuring Source IP Lockdown

To configure source IP lockdown, you must enable *DHCP* snooping on the ports connected to the DHCP server and DHCP client before you enable source IP lockdown.

You must enable source IP lockdown on the ports connected to the DHCP client, not on the ports connected to the DHCP server.

Note
Enabling DHCP snooping and source IP lockdown on the same port applies ACL rules with the same match conditions, but different actions. The rule with deny action takes precedence, so packets are dropped if the these ACL rules are installed on different slices. Many factors influence which slice rules are installed on. To see which slice these rules are installed on, use the command `show access-list usage acl-slice port port` or `show access-list usage acl-rule port port` .

- Enable DHCP snooping using the command:
  `enable ip-security dhcp-snooping [dynamic | {vlan} vlan_name] ports [all | ports] violation-action [drop-packet {[block-mac | block-port] [duration duration_in_seconds | permanently] | none}] {snmp-trap}`
  For more information about DHCP snooping see, Configuring DHCP Snooping on page 1567.
  Source IP lockdown is disabled on the switch by default.
- To enable source IP lockdown, use the command:
  `enable ip-security source-ip-lockdown ports [all | ports]`
- To disable source IP lockdown, use the command
  `disable ip-security source-ip-lockdown ports [all | ports]`

Displaying Source IP Lockdown Information

- Display the source IP lockdown configuration on the switch using the ocmmand:.
  `show ip-security source-ip-lockdown`
  Below is sample output from this command:

```
Ports           Locked IP Address
23              10.0.0.101
```

Clear Source IP Lockdown Information

- Remove existing source IP lockdown entries on the switch using the command:
  `clear ip-security source-ip-lockdown entries ports [ports | all]`

ARP Learning

The address resolution protocol (ARP) is part of the TCP/IP suite used to dynamically associate a device's physical address (MAC address) with its logical address (IP address).

The switch broadcasts an ARP request that contains the IP address, and the device with that IP address sends back its MAC address so that traffic can be transmitted across the network. The switch maintains an ARP table (also known as an ARP cache) that displays each MAC address and its corresponding IP address.

By default, the switch builds its ARP table by tracking ARP requests and replies, which is known as ARP learning. You can disable ARP learning so that the only entries in

the ARP table are either manually added or those created by *DHCP* secured ARP; the switch does not add entries by tracking ARP requests and replies. By disabling ARP learning and adding a permanent entry or configuring DHCP secured ARP, you can centrally manage and allocate client IP addresses and prevent duplicate IP addresses from interrupting network operation.

Configuring ARP Learning

As previously described, ARP learning is enabled by default. The switch builds its ARP table by tracking ARP requests and replies.

- Disable ARP learning on one or more ports in a *VLAN* using the command:

  `disable ip-security arp learning learn-from-arp [dynamic | {vlan} vlan_name] ports [all | ports]`

- Re-enable ARP learning on one or more ports in a VLAN using the command:

  `enable ip-security arp learning learn-from-arp [dynamic | {vlan} vlan_name] ports [all | ports]`

Adding a Permanent Entry to the ARP Table

If you disable ARP learning, you must either manually add a permanent entry to the ARP table or configure *DHCP* secured ARP to populate the ARP table.

- Manually add a permanent entry to the ARP table using the command:

  `configure iparp add ip_addr {vr vr_name} mac`

For more detailed information about this command and IP routing, see IPv4 Unicast Routing.

Configuring DHCP Secured ARP

Before you configure *DHCP* secured ARP, you must enable DHCP snooping on the switch.

Another method available to populate the ARP table is DHCP secured ARP. DHCP secured ARP requires that ARP entries be added to or deleted from the ARP table only when the DHCP server assigns or re-assigns an IP address. These entries are known as a secure ARP entry. If configured, the switch adds the MAC address and its corresponding IP address to the ARP table as a permanent ARP entry. Regardless of other ARP requests and replies seen by the switch, the switch does not update secure ARP entries. DHCP secured ARP is linked to the “DHCP snooping” feature. The same DHCP bindings database created when you enabled DHCP snooping is also used by DHCP secured ARP to create secure ARP entries. The switch only removes secure

ARP entries when the corresponding DHCP entry is removed from the trusted DHCP bindings database.

Note
If you enable DHCP secured ARP on the switch without disabling ARP learning, ARP learning continues which allows insecure entries to be added to the ARP table.

- Enable DHCP snooping using the command:

  `enable ip-security dhcp-snooping [dynamic | {vlan} vlan_name] ports [all | ports] violation-action [drop-packet {[block-mac | block-port] [duration duration_in_seconds | permanently] | none}] {snmp-trap}`

  For more information about DHCP snooping see, Configuring DHCP Snooping on page 1567.

  DHCP secured ARP learning is disabled by default.
- Enable DHCP secured ARP using the command:

  `enable ip-security arp learning learn-from-dhcp [dynamic vlan | {vlan} vlan_name] ports [all | ports]`

  DHCP Secured ARP must be enabled on the DHCP server port as well as the DHCP client ports.
- Disable DHCP secured ARP using the command:

  `disable ip-security arp learning learn-from-dhcp [dynamic vlan | {vlan} vlan_name ports [all | ports]`

Note
You must enable DHCP secured ARP on the DHCP server, as well as on the client ports. DHCP snooping must also be enabled on both the server and client ports.

Displaying ARP Information

- Display how the switch builds an ARP table and learns MAC addresses for devices on a specific *VLAN* and associated member ports using the command:.

  `show ip-security arp learning {vlan} vlan_name | vlan_list`

  Below is sample output from this command:

```
Port                Learn-from
----------------------------------
2:1                 ARP
2:2                 DHCP
2:3                 ARP
2:4                 None
2:5                 ARP
2:6                 ARP
2:7                 ARP
2:8                 ARP
```

- View the ARP table, including permanent and *DHCP* secured ARP entries using the command:

  ```
  show iparp {ip_address | mac | vlan {vlan_name | vlan_list} |
  permanent} {vr vr_name}
  ```

  Note
  DHCP secured ARP entries are stored as static entries in the ARP table.

Gratuitous ARP Protection

When a host sends an ARP request to resolve its own IP address it is called *gratuitous ARP*. A gratuitous ARP request is sent with the following parameters:

- Destination MAC address—FF:FF:FF:FF:FF:FF (broadcast)
- Source MAC address—Host's MAC address
- Source IP address = Destination IP address—IP address to be resolved

In a network, gratuitous ARP is used to:

- Detect duplicate IP address.

  In a properly configured network, there is no ARP reply for a gratuitous ARP request. However, if another host in the network is configured with the same IP address as the source host, then the source host receives an ARP reply.
- Announce that an IP address has moved or bonded to a new network interface card (NIC).

  If you change a system NIC, the MAC address to its IP address mapping also changes. When you reboot the host, it sends an ARP request packet for its own IP address. All of the hosts in the network receive and process this packet. Each host updates their old mapping in the ARP table with this new mapping
- Notify a Layer 2 switch that a host has moved from one port to another port.

However, hosts can launch man-in-the-middle attacks by sending out gratuitous ARP requests for the router's IP address. This results in hosts sending their router traffic to the attacker, and the attacker forwarding that data to the router. This allows passwords, keys, and other information to be intercepted.

To protect against this type of attack, the router sends out its own gratuitous ARP request to override the attacker whenever a gratuitous ARP request broadcast packet with the router's IP address as the source is received on the network.

If you enable both *DHCP* secured ARP and gratuitous ARP protection, the switch protects its own IP address and those of the hosts that appear as secure entries in the ARP table.

Configuring Gratuitous ARP

You enable the gratuitous ARP feature on a per *VLAN* basis, not on a per port basis. The validation is done for all gratuitous ARP packets received on a VLAN in which this feature is enabled irrespective of the port in which the packet is received.

When enabled, the switch generates gratuitous ARP packets when it receives a gratuitous ARP request where either of the following is true:

- The sender IP is the same as the switch VLAN IP address and the sender MAC address is not the switch MAC address.
- The sender IP is the same as the IP of a static entry in the ARP table and the sender MAC address is not the static entry's MAC address.

When the switch generates an ARP packet, the switch generates logs and traps.

- Enable gratuitous ARP protection using the command:

  ```
  enable ip-security arp gratuitous-protection [dynamic | {vlan} all | vlan_name]
  ```

- In addition, to protect the IP addresses of the hosts that appear as secure entries in the ARP table, use the following commands to enable *DHCP* snooping, DHCP secured ARP, and gratuitous ARP on the switch:

  ```
  enable ip-security dhcp-snooping [dynamic | {vlan} vlan_name] ports [all | ports] violation-action [drop-packet {[block-mac | block-port] [duration duration_in_seconds | permanently] | none}] {snmp-trap}
  ```

  ```
  enable ip-security arp learning learn-from-dhcp {vlan} vlan_name ports [all | ports]
  ```

  ```
  enable ip-security arp gratuitous-protection {vlan} [all | vlan_name]
  ```

- Disable gratuitous ARP protection using the command:

  ```
  disable ip-security arp gratuitous-protection [dynamic | {vlan} vlan_name |all ]
  ```

- In ExtremeXOS 11.5 and earlier, you enable gratuitous ARP protection using the following command:

  ```
  enable iparp gratuitous protect vlan vlan-name
  ```

- In ExtremeXOS11.5 and earlier, you disable gratuitous ARP protection with the following command:

  ```
  disable iparp gratuitous protect vlan vlan-name
  ```

Displaying Gratuitous ARP Information

- Display information about gratuitous ARP using the command:

  ```
  show ip-security arp gratuitous-protection
  ```

  Below is sample output from this command:

  ```
  Gratuitous ARP Protection enabled on following VLANs:
  Default, test
  ```

ARP Validation

ARP validation is also linked to the "*DHCP* snooping" feature. The same DHCP bindings database created when you enabled DHCP snooping is also used to validate ARP entries arriving on the specified ports.

Validation Option	ARP Request Packet Type	ARP Response Packet Type
DHCP		Source IP is not present in the DHCP snooping database OR is present but Source Hardware Address doesn't match the MAC in the DHCP bindings entry.
IP	Source IP == Mcast OR Target IP == Mcast OR Source IP is not present in the DHCP snooping database OR Source IP exists in the DHCP bindings database but Source Hardware Address doesn't match the MAC in the DHCP bindings entry.	Source IP == Mcast OR Target IP == Mcast
Source-MAC	Ethernet source MAC does not match the Source Hardware Address.	Ethernet source MAC does not match the Source Hardware Address.
Destination-MAC		Ethernet destination MAC does not match the Target Hardware Address.

Depending on the options specified when enabling ARP validation, the following validations are done. Note that the 'DHCP' option does not have to be specified explicitly, it is always implied when ARP validation is enabled.

Configuring ARP Validation

Before you configure ARP validation, you must enable *DHCP* snooping on the switch.

- Enable DHCP snooping using the command:.
  `enable ip-security dhcp-snooping [dynamic | {vlan} vlan_name] ports [all | ports] violation-action [drop-packet {[block-mac | block-port] [duration duration_in_seconds | permanently] | none}] {snmp-trap}`

  For more information about DHCP snooping see, Configuring DHCP Snooping on page 1567.

  ARP validation is disabled by default.
- Enable and configure ARP validation using the command:
  `enable ip-security arp validation {destination-mac} {source-mac} {ip} [dynamic vlan_id |{vlan} vlan_name] [all | ports] violation-action [drop-packet {[block-port] [duration duration_in_seconds | permanently]}] {snmp-trap}`

  The violation action setting determines what action(s) the switch takes when an invalid ARP is received. Any violation that occurs causes the switch to generate an

EMS log message. You can configure to suppress the log messages by configuring EMS log filters. For more information about EMS, see the section Using the Event Management System/Logging on page 1750.

- Disable ARP validation using the command:
  `disable ip-security arp validation [dynamic | {vlan} vlan_name] [all | ports]`

Displaying ARP Validation Information

- Display information about ARP validation using the command:
  `show ip-security arp validation {vlan} vlan_name`

  Below is sample output from this command:

```
-------------------------------------------------------------------
Port    Validation              Violation-action
-------------------------------------------------------------------
7       DHCP                    drop-packet, block-port for 120 seconds, snmp-trap
23      DHCP                    drop-packet, block-port for 120 seconds, snmp-trap
```

VSA RADIUS Support for DHCP Snooping and ARP Validation

RADIUS support for *DHCP* DHCP snooping and ARP validation allows users transitioning from Ethernet Routing Switch to ExtremeXOS at Edge with Fabric Attach using RADIUS to dynamically configure port security through RADIUS VSA Attributes. ExtremeXOS can provision items via RADIUS VSA Attributes currently supported with ERS switching. Dynamic configuration can be applied to the following VLAN settings:

- The ability to enable Dynamic ARP Inspection.
- The ability to enable DHCP Snooping.

The RADIUS user configuration attributes for VLAN settings can specify a single VLAN or a range of VLANs for each setting request. The RADIUS user configuration attributes, which request the settings, include:

- Fabric-Attach-Service-Request += "DAI:<vid>[-<vid>]"
- Fabric-Attach-Service-Request += "DHCPSNOOP:<vid>[-<vid>]"

This FA-Service-Request attribute contains a DHCP Snoop and ARP Validation enable status for VLAN. When a new user is authenticated by netlogin, the new attributes of the DHCP Snoop and ARP Validation enable is given to netlogin. Netlogin will process and send the attribute message to the IP Security module as "enable". If all users are unauthenticated by netlogin, a disable message for the VLAN will be sent to IP Security. RADIUS configuration is applicable for both static and dynamic VLANs.

Whenever a DHCP Snooping or ARP Validation configuration is received from RADIUS, IP Security will enable the feature on all the ports in VLAN with the violation action "Drop-packet."

RADIUS Configuration Over ExtremeXOS CLI

RADIUS configuration takes precedence over ExtremeXOS CLI. Once RADIUS configuration is applied for a VLAN, any CLI command action (for example disable DHCP snoop for a port or changing violation action) will be ignored, but the configuration will be saved. Once the RADIUS configuration is disabled, CLI

configuration will be applied. The following tables summarize the various DHCP snoop configuration actions with respect to ExtremeXOS CLI and RADIUS configurations. The same configuration actions are also applicable for ARP Validation.

The following command equivalencies exist for RADIUS configurations received for a VLAN:

DHCPSNOOP:<vid> is equivalent to `enable ip-security dhcp-snooping vlan default ports all violation-action drop-packet`

DAI:<vid> is equivalent to `enable ip-security arp validation Default ports all violation-action drop-packet`

Table 141: EXOS CLI Configuration over RADIUS Configuration Actions

ExtremeXOS CLI configuration	RADIUS configuration	Configuration action
Enable DHCP snooping for a VLAN and all VLAN ports with the violation action “None"	Enable DHCP snooping for a VLAN	DHCP snooping will be enabled for all VLAN ports with the violation action “Drop packet”
Enable DHCP snooping for a VLAN but not all ports	Enable DHCP snooping for a VLAN	DHCP snooping will be enabled for all VLAN ports with the violation action “Drop packet”
Disable DHCP snooping for a VLAN and all its ports	Enable DHCP snooping for a VLAN	DHCP snooping will be enabled for all VLAN ports with the violation action “Drop packet”

Table 142: RADIUS Configuration over EXOS CLI Configuration Actions

RADIUS configuration	ExtremeXOS CLI Configuration	Configuration action
Enable DHCP snooping for a VLAN	Disable DHCP snooping for a VLAN and all its port.	DHCP snooping will be enabled for all VLAN ports with violation action as "Drop packet".
Enable DHCP snooping for a VLAN	Modify DHCP snoop violation action for VLAN to “port-block permanently”	DHCP snooping will be enabled for all VLAN ports with violation action as "Drop packet".

DHCP Security Support for MLAG Controlling Bridge

DHCP Security Support for MLAG Controlling Bridge implements the checkpointing of DHCP snooping entries and DHCP and ARP violations between MLAG peers. This allows

IP security to share the data or state required for its features, and to exhibit identical behavior in the MLAG topology.

Note
DHCP security for MLAG controlling bridges is also supported in VPEX and W-MLAG topologies.

In a typical deployment, a DHCP snooping entry is captured in MLAG Peer1 when it receives a DHCP ACK packet from the DHCP server and forwards it to the DHCP client. The DHCP snoop entry is then checkpointed to MLAG Peer2 as soon as it is received by the IP security module. This ensures redundancy and identical functional behavior in MLAG Peer2 when MLAG Peer1 goes down for any reason:

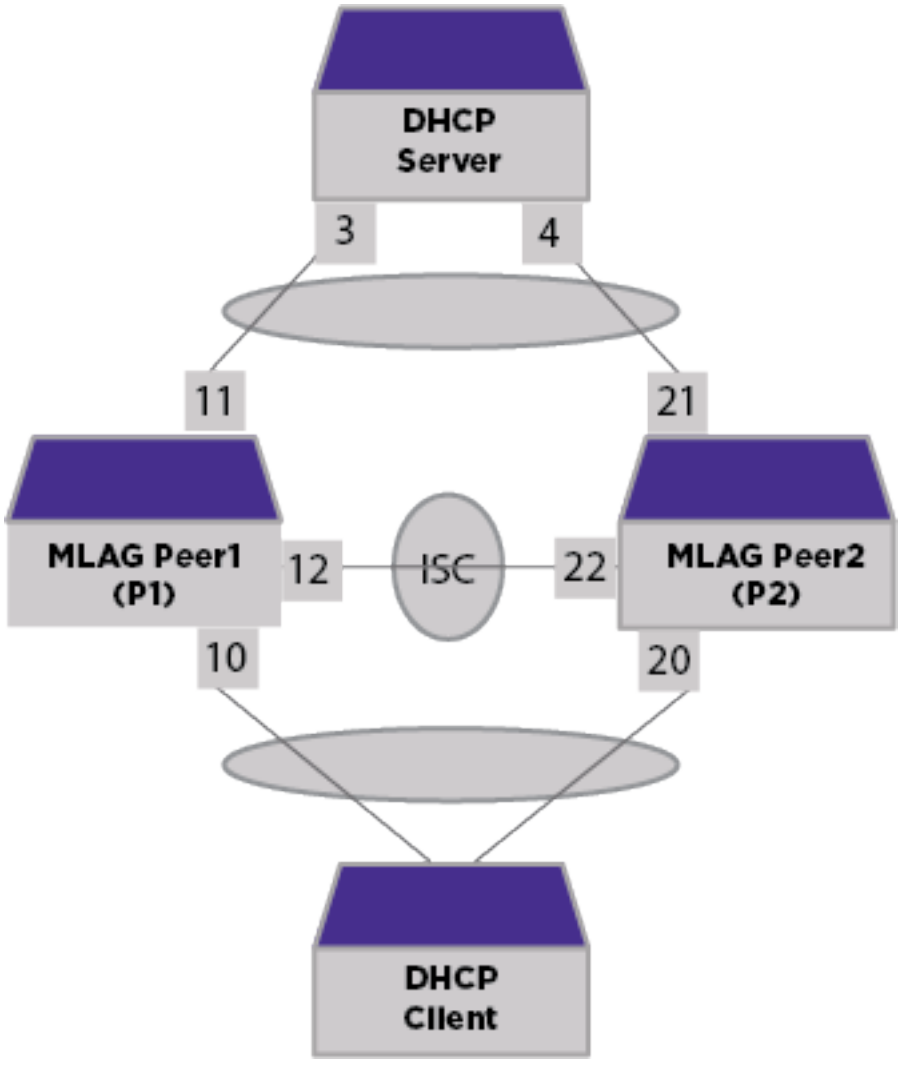


Figure 194: MLAG Topology

In the following configuration, if a DHCP entry is snooped by Peer1 and checkpointed to Peer2, Peer2 must allow ARP requests arriving on its Port 1 with an IP address that exists in the DHCP snooping database:

MLAG Peer1 configuration

```
create vlan v1 tag 900
configure vlan v1 add port 10,11,12 tagged
enable ip-security dhcp-snooping vlan v1 ports 11 violation-action drop-packet block-port
duration 300
enable ip-security arp learning learn-from-dhcp vlan v1 ports 11
enable ip-security arp validation v1 ports 11 violation-action drop-packet block-port
duration 300
```

MLAG Peer2 configuration

```
create vlan v1 tag 900
configure vlan v1 add port 20,21,22 tagged
enable ip-security dhcp-snooping vlan v1 ports 21 violation-action drop-packet block-port
duration 300
enable ip-security arp learning learn-from-dhcp vlan v1 ports 21
enable ip-security arp validation v1 ports 21 violation-action drop-packet block-port
duration 300
```

MLAG Checkpointing

After Inter-Switch Connection (ISC) connectivity between the MLAG peers is established, IP security begins checkpointing DHCP snooping and DHCP or ARP violation actions. As long as MLAG peers have ISC connectivity, addition and deletion of entries/data will be checkpointed. The checkpoint is only done for MLAG ports.

DHCP Snoop Checkpointing

When a DHCP snoop entry is checkpointed from Peer1 to Peer2, the receiving peer checks if it has the received DHCP snoop entry present in its DHCP bindings table. If it is a new entry, it will be added to the DHCP binding table with the server ports as ISC ports. The client port is then identified using FDB lookup on the received client MAC. If the receiving entry already exists, the lease time will be updated so that it is in sync with the peer. If the server ports are BPE ports, the checkpoint receiving peer will set to the BPE port as a server instead of an ISC port.

Using the previous configuration example, once a DHCP ACK is received, Peer1 creates a new DHCP snoop entry into its DHCP binding table, with the server port as 11, the client port as 10, and checkpoints the entry to Peer2. On receiving the checkpoint entry, Peer2 adds the entry to its DHCP binding table with the server port as 22 (ISC port) and the client port as 20 (FDB lookup port).

DHCP and ARP Violation Checkpoint

DHCP and ARP Violation events - “block-port” and “block-mac” are checkpointed between MLAG peers for front panel ports. However, violation actions are not applied at the receiving peer. The receiving peer stores the block action, block duration internally, and applies the checkpointed block time (violationTime - lapsedTime) once a similar violation occurs. For example, if Peer1 receives a MAC violation on MAC “00:04:96:97:fa:a2”, then IP security will block the MAC for 300 seconds and checkpoint the information to Peer2. Once the same MAC “00:04:96:97:fa:a2” violation is seen in Peer2 after a period of time, Peer2 will block the MAC for remaining 200 seconds

(instead of 300 seconds). This ensures both Peer1 and Peer2 removes or unblocks the violation action at the same time.

Tip
With BPE ports, the violation action is also checkpointed and applied in the peer.

Denial of Service Protection

A Denial-of-Service (DoS) attack occurs when a critical network or computing resource is overwhelmed and rendered inoperative in a way that legitimate requests for service cannot succeed.

In its simplest form, a DoS attack is indistinguishable from normal heavy traffic. There are some operations in any switch or router that are more costly than others, and although normal traffic is not a problem, exception traffic must be handled by the switch's CPU in software.

Note
DoS protection is not supported for IPv6.

Some packets that the switch processes in the CPU software include:

- Traffic resulting from new MAC learning.

  Note
  When certain features such as Network Login are enabled, hardware learning is disabled to let software control new MAC learning.

- Routing and control protocols including *ICMP*, *BGP (Border Gateway Protocol)*, *OSPF (Open Shortest Path First)*, *STP*, *EAPS (Extreme Automatic Protection Switching)*, *ESRP*, etc.
- Switch management traffic (switch access by Telnet, SSH, HTTP, *SNMP*, etc.)
- Other packets directed to the switch that must be discarded by the CPU

If any one of these functions is overwhelmed, the CPU may be too busy to service other functions and switch performance will suffer. Even with very fast CPUs, there will always be ways to overwhelm the CPU with packets that require costly processing.

DoS Protection is designed to help prevent this degraded performance by attempting to characterize the problem and filter out the offending traffic so that other functions can continue. When a flood of CPU bound packets reach the switch, DoS Protection will count these packets. When the packet count nears the alert threshold, packets headers are saved. If the threshold is reached, then these headers are analyzed, and a hardware *ACL* is created to limit the flow of these packets to the CPU. This ACL will remain in place to provide relief to the CPU. Periodically, the ACL will expire, and if the attack is

still occurring, it will be re-enabled. With the ACL in place, the CPU will have the cycles to process legitimate traffic and continue other services.

Note
User-created ACLs take precedence over the automatically applied DoS protect ACLs.

DoS Protection sends a notification when the notify threshold is reached.

You can also specify some ports as trusted ports, so that DoS protection is not applied to those ports.

Configuring Simulated Denial of Service Protection

The conservative way to deploy DoS protection is to use the simulated mode first. In simulated mode, DoS protection is enabled, but no *ACL*s are generated.

- Enable the simulated mode using the command:
  `enable dos-protect simulated`

  This mode is useful to gather information about normal traffic levels on the switch. This will assist in configuring denial of service protection so that legitimate traffic is not blocked.

  The following topics describe how to configure DoS protection, including alert thresholds, notify thresholds, ACL expiration time, and so on.

Configuring Denial of Service Protection

- Enable or disable DoS protection using the command:
  `enable dos-protect`
  `disable dos-protect`

  After enabling DoS protection, the switch will count the packets handled by the CPU and periodically evaluate whether to send a notification and/or create an *ACL* to block offending traffic.

  You can configure a number of the values used by DoS protection if the default values are not appropriate for your situation.

  The values that you can configure are:

  - interval—How often, in seconds, the switch evaluates the DoS counter (default: 1 second)
  - alert threshold—The number of packets received in an interval that will generate an ACL (default: 4000 packets)
  - notify threshold—The number of packets received in an interval that will generate a notice (default: 3500 packets)
  - ACL expiration time—The amount of time, in seconds, that the ACL will remain in place (default: 5 seconds)
- Configure the interval at which the switch checks for DoS attacks using the command:
  `configure dos-protect interval` *`seconds`*

- Configure the alert threshold using the command:
  `configure dos-protect type l3-protect alert-threshold` *packets*
- Configure the notification threshold using the command:
  `configure dos-protect type l3-protect notify-threshold` *packets*
- Configure the ACL expiration time using the command:
  `configure dos-protect acl-expire` *seconds*

Configuring Trusted Ports

Traffic from trusted ports will be ignored when DoS protect counts the packets to the CPU. If we know that a machine connected to a certain port on the switch is a safe "trusted" machine, and we know that we will not get a DoS attack from that machine, the port where this machine is connected to can be configured as a trusted port, even though a large amount of traffic is going through this port.

- Configure the trusted ports list using the command:
  `configure dos-protect` **`trusted-ports`** [**`ports`** [*ports* | **`all`**] | **`add-ports`** [*ports-to-add* | **`all`**] | **`delete-ports`** [*ports-to-delete* | **`all`**]]

Displaying DoS Protection Settings

- Display the DoS protection settings using the command:
  `show dos-protect {`**`detail`**`}`

Protocol Anomaly Protection

The Extreme chipsets contain built-in hardware protocol checkers that support port security features for security applications, such as stateless DoS protection.

The protocol checkers allow users to drop the packets based on the following conditions, which are checked for ingress packets prior to the L2/L3 entry table:

- SIP = DIP for IPv4/IPv6 packets.
- TCP_SYN Flag = 0 for IPv4/IPv6 packets
- TCP Packets with control flags = 0 and sequence number = 0 for IPv4/IPv6 packets
- TCP Packets with FIN, URG & PSH bits set & seq. number = 0 for IPv4/IPv6 packets
- TCP Packets with SYN & FIN bits are set for IPv4/IPv6 packets
- TCP Source Port number = TCP Destination Port number for IPv4/IPv6 packets
- First TCP fragment does not have the full TCP header (less than 20 bytes) for IPv4/IPv6 packets
- TCP header has fragment offset value as 1 for IPv4/IPv6 packets
- UDP Source Port number = UDP Destination Port number for IPv4/IPv6 packets
- ICMP ping packets payload is larger than programmed value of *ICMP*max size for IPv4/IPv6 packets
- Fragmented ICMP packets for IPv4/IPv6 packets

The protocol anomaly detection security functionality is supported by a set of anomaly-protection enable, disable, configure, clear, and show CLI commands. For further details, see the Switch Engine v33.6.1 Command References .

Flood Rate Limitation

Flood rate limitation, or storm control, is used to minimize the network impact of ingress flooding traffic. You can configure ports to accept a specified rate of packets per second. When that rate is exceeded, the port blocks traffic and drops subsequent packets until the traffic again drops below the configured rate.

- Configure the rate limit.

  `configure ports port_list rate-limit flood [broadcast | multicast | unknown-destmac] [no-limit | pps]`

- Display rate limiting statistics.

  `show ports {port_list} rate-limit flood {no-refresh}`

  Note
  ExtremeXOS switches implement rate limiting granularity at millisecond intervals. The traffic bursts are monitored at millisecond intervals and the actions are performed within sub-seconds (when applicable). When the switch evaluates the traffic pattern for bursts against the configured value in pps, the value is calibrated on a per-millisecond interval. For example, using the `configure port 1 rate-limit flood broadcast1000` command would be equivalent to 1 packet per millisecond.

- Configure ExtremeCloud IQ Agent logging behavior.

  The ExtremeCloud IQ Agent logging feature uses in-memory, buffer-based logging as the default mechanism instead of continuous on-disk file logging. This reduces storage consumption while maintaining diagnostic capabilities.

  In-memory buffer logs for both the Python ExtremeCloud IQ Agent application and the HiveAgent process are stored in EMS trace buffers and are included in `show tech-support` output for support escalations and diagnostics.

  File-based, on-disk logging is disabled by default and available only as an on-demand facility for extended troubleshooting sessions when detailed logging is required.

  a. View ExtremeCloud IQ Agent buffer logs.

     ExtremeCloud IQ Agent buffer logs are captured in real-time and included in system diagnostics:

     ```
     show tech-support
     ```

     The output includes recent activity from both the Python application and HiveAgent process buffers, eliminating the need for persistent on-disk logging during normal operations.

  b. Enable file-based logging on demand for troubleshooting.

     For targeted troubleshooting sessions, file-based logging can be enabled using debug CLI. This is configured using debug commands at critical, error, warning, info, and debug severity levels.

     For detailed configuration procedures and complete CLI command syntax for enabling file-based logging, refer to the user guide.

c. Note behavior change after upgrade.

Upgrade Note: After upgrading to version 33.7.1, ExtremeCloud IQ Agent file-based, on-disk logging is off by default. The switch does not write new entries to on-disk log files. Any existing log files present on the switch at the time of upgrade are preserved and are not deleted or truncated. Recent activity remains available through the in-memory buffer-based trace feature.

To resume or extend on-disk file-based logging during a troubleshooting session, enable file-based logging using the debug CLI. Disabling file-based logging stops new writes but preserves previously captured on-disk log files.

Authenticating Management Sessions Through a TACACS+ Server

You can use a Terminal Access Controller Access Control System Plus (TACACS+) server to authenticate management sessions for multiple switches.

A TACACS+ server allows you to centralize the authentication database, so that you do not have to maintain a separate local database on each switch. TACACS+ servers provide the following services:

- Username and password authentication
- Command authorization (the TACACS+ server validates whether the user is authorized to execute each command within the subset of commands, based on login privilege level)

  Note
  Command usage that should be restricted for a user account by TACACS with CLI authorization enabled may not occur when users are logged in by Chalet or when using the XML API directly.

- Accounting service (tracks authentication and authorization events)

  Note
  You can use a local database on each switch as a backup authentication service if the TACACS+ service is unavailable. When the TACACS+ service is operating, privileges defined on the TACACS+ server take precedence over privileges configured in the local database.

To use TACACS+ server features, you need the following components:

- TACACS+ client software, which is included in the ExtremeXOS software.

- A TACACS+ server, which is a third-party product.

Note

- TACACS+ provides many of the same features provided by *RADIUS (Remote Authentication Dial In User Service)*, but enabling both RADIUS and TACACS+ at the same time is not supported for Management User Authentication.
- RADIUS can be used for both Switch Management User Authentication as well as Network Login user/device authentication, while TACACS+ can be used only for Management User Authentication.

TACACS+ is a communications protocol that is used between client and server to implement the TACACS+ service. The TACACS+ client component of the ExtremeXOS software should be compatible with any TACACS+ compliant server product.

Note
The switch allows local authentication when the client IP is excluded in TACACS+ server by default. To disallow local authentication when the client IP is excluded in TACACS+ server the local authentication disallow option should be used.

Note
Version 32.6 adds support for configuration of TACACS+ IPv6 server.

For information on installing, configuring, and managing a TACACS+ server, see the product documentation for that server.

The following describes how to configure the ExtremeXOS TACACS+ client component in the ExtremeXOS software: Configuring the TACACS+ Client for Authentication and Authorization on page 1591.

Configuring the TACACS+ Client for Authentication and Authorization

Changing the TACACS+ Server

Use the following steps to change TACACS+ server configuration to avoid service interruption of authentication and authorization.

1. Unconfigure the existing primary TACACS+ server.

   Note
   The command `disable tacacs` is *not* required while changing TACACS+ servers.

   If only a single TACACS+ server is configured, you must disable TACACS authorization (if enabled) before reconfiguring TACACS+ server:

   ```
   disable tacacs-authorization
   ```

   To unconfigure the existing primary TACACS+ server:

```
unconfigure tacacs server [primary | secondary]
```

Note
After this step, TACACS+ will failover to secondary server.

2. Configure the new primary TACACS+ server:

```
configure tacacs [primary | secondary] server [host_ipaddr |
host_ipV6addr | hostname] {tcp_port} client-ip [client_ipaddr |
client_ipV6addr] {vr vr_name}
```

3. Configure the shared-secret password for primary TACACS+ server:

```
configure tacacs [primary | secondary] shared-secret {encrypted}
string
```

Note
Only after configuring shared-secret password for primary server, TACACS+ will fallback to primary server from secondary.

4. Unconfigure existing secondary TACACS+ server:

```
unconfigure tacacs server [primary | secondary]
```

5. Configure new secondary TACACS+ server:

```
configure tacacs [primary | secondary] server [host_ipaddr |
host_ipV6addr | hostname] {tcp_port} client-ip [client_ipaddr |
client_ipV6addr] {vr vr_name}
```

6. Configure shared-secret password for secondary TACACS+ server:

```
configure tacacs [primary | secondary] shared-secret {encrypted}
string
```

Specifying TACACS+ Server Addresses

Before the TACACS+ client software can communicate with a TACACS+ server, you must specify the server address in the client software. You can specify up to two TACACS+ servers, and you can use either an IP address or a host name to identify each server.

- To configure the TACACS+ servers in the client software, use the following command:

  ```
  configure tacacs [primary | secondary] server [host_ipaddr |
  host_ipV6addr | hostname] {tcp_port} client-ip ipaddress {vr vr_name}
  ```

  To configure the primary TACACS+ server, specify primary. To configure the secondary TACACS+ server, specify secondary.

Configuring the TACACS+ Client Timeout Value

- To configure the timeout if a server fails to respond, use the following command:

  ```
  configure tacacs timeout seconds
  ```

  To detect and recover from a TACACS+ server failure when the timeout has expired, the switch makes one authentication attempt before trying the next designated TACACS+ server or reverting to the local database for authentication. In the event that the switch still has IP connectivity to the TACACS+ server, but a TCP session

cannot be established, (such as a failed TACACS+ daemon on the server), fail over happens immediately regardless of the configured timeout value.

For example, if the timeout value is set for three seconds (the default value), it will take three seconds to fail over from the primary TACACS+ server to the secondary TACACS+ server. If both the primary and the secondary servers fail or are unavailable, it takes approximately six seconds to revert to the local database for authentication.

Configuring the Shared Secret Password for TACACS+ Communications

The shared secret is a password that is configured on each network device and TACACS+ server. The shared secret is used to verify communication between network devices and the server.

- To configure the shared secret for client communications with TACACS+ servers, use the following command:

  `configure tacacs [primary | secondary] shared-secret {encrypted} string`

  To configure the shared secret for a primary TACACS+ server, specify primary. To configure the shared secret for a secondary TACACS+ server, specify secondary.

  Do not use the encrypted keyword to set the shared secret. The encrypted keyword prevents the display of the shared secret in the `show configuration` command output.

Enabling and Disabling the TACACS+ Client Service

The TACACS+ client service can be enabled or disabled without affecting the client configuration. When the client service is disabled, the client does not communicate with the TACACS+ server, so authentication must take place through the another service such as the local database or a *RADIUS* server.

Note
You cannot use RADIUS and TACACS+ at the same time.

- To enable the TACACS+ client service, use the following command: `enable tacacs`
- To disable the TACACS+ client service, use the following command: `disable tacacs`

Requiring the Privilege Level Attribute for Authentication

You can require that the privilege level attribute (priv-lvl) must be specified for TACACS authentication to occur. Setting the privilege level attribute as required does not change any behavior associated with values received in the priv-lvl attribute, only the presence/absence of the attribute.

To require that the priv-lvl is set to allow authentication, use the following command:

`configure tacacs priv-lvl [required | optional]`

TACACS+ Configuration Example

This section provides a sample TACACS+ client configuration that:

- Specifies the primary TACACS+ server.
- Specifies the shared secret for the primary TACACS+ server.
- Specifies the secondary TACACS+ server.
- Specifies the shared secret for the secondary TACACS+ server.
- Enables the TACACS+ client on the switch.
- Configures an IPv6 server as the primary TACACS+ server.

All other client configuration features use the default settings as described Configuring the TACACS+ Client for Authentication and Authorization on page 1591 or in the Switch Engine v33.6.1 Command References .

```
configure tacacs primary server 10.201.31.238 client-ip 10.201.31.85 vr "VR-Default"
configure tacacs primary shared-secret purple
configure tacacs secondary server 10.201.31.235 client-ip 10.201.31.85 vr "VR-Default"
configure tacacs secondary shared-secret purple
enable tacacs
```

IPv6 servers can be configured similarly using the following commands:

```
configure tacacs primary server 2111::220 client-ip 2111::85 vr "VR-Default"
configure tacacs primary shared-secret purple
configure tacacs secondary server 2111::210 client-ip 2111::85 vr "VR-Default"
configure tacacs secondary shared-secret purple
configure tacacs primary server 1111::220 client-ip 1111::170 vr vr-mgmt
```

To display the TACACS+ client configuration, use the `show tacacs` command. Below is sample output from this command:

```
TACACS+: enabled
TACACS+ Authorization: disabled
TACACS+ Accounting : disabled
TACACS+ Server Connect Timeout sec: 3
Primary TACACS+ Server:
    Server name   :
    IP address    :  10.201.31.238
    Server IP Port:  49
    Client address:  10.201.31.85 (VR-Default)
    Shared secret :  purple
Secondary TACACS+ Server:
    Server name   :
    IP address    :  10.201.31.235
    Server IP Port:  49
    Client address:  10.201.31.85 (VR-Default)
    Shared secret :  purple
TACACS+ Acct Server Connect Timeout sec: 3
Primary TACACS+ Accounting Server:Not configured
Secondary TACACS+ Accounting Server:Not configured
```

Configuring the TACACS+ Client for Accounting

Specifying the Accounting Server Addresses

Before the TACACS+ client software can communicate with a TACACS+ accounting server, you must specify the server address in the client software. You can specify up to

two accounting servers, and you can use either an IP address or a host name to identify each server.

- To specify TACACS+ accounting servers, use the following command:
  `configure tacacs-accounting [primary | secondary] server [host_ipaddr | host_ipV6addr | hostname] {tcp_port} client-ip [client_ipaddr | client_ipVaddr] {vr vr_name}`

  To configure the primary TACACS+ accounting server, specify primary. To configure the secondary TACACS+ accounting server, specify secondary.

Configuring the TACACS+ Client Accounting Timeout Value

- To configure the timeout if a server fails to respond, use the following command:
  `configure tacacs-accounting timeout seconds`

  To detect and recover from a TACACS+ accounting server failure when the timeout has expired, the switch makes one authentication attempt before trying the next designated TACACS+ accounting server or reverting to the local database for authentication. In the event that the switch still has IP connectivity to the TACACS+ accounting server, but a TCP session cannot be established, (such as a failed TACACS+ daemon on the accounting server), failover happens immediately regardless of the configured timeout value. If the user does not have a local account or the user is disabled locally, the user's login will fail.

  For example, if the timeout value is set for three seconds (the default value), it takes three seconds to failover from the primary TACACS+ accounting server to the secondary TACACS+ accounting server. If both the primary and the secondary servers fail or are unavailable, it takes approximately six seconds to revert to the local database for authentication.

Configuring the Shared Secret Password for TACACS+ Accounting Servers

The shared secret is a password that is configured on each network device and TACACS+ accounting server. The shared secret is used to verify communication between network devices and the server.

- To configure the shared secret for client communications with TACACS+ accounting servers, use the following command:
  `configure tacacs-accounting [primary | secondary] shared-secret {encrypted} string`

  To configure the primary TACACS+ accounting server, specify primary. To configure the secondary TACACS+ accounting server, specify secondary.

  Do not use the encrypted keyword to set the shared secret. The encrypted keyword prevents the display of the shared secret in the `show configuration` command output.

Enabling and Disabling TACACS+ Accounting

After you configure the TACACS+ client with the TACACS+ accounting server information, you must enable accounting in the TACACS+ client before the switch begins transmitting the information. You must enable TACACS+ authentication in

the client for accounting information to be generated. You can enable and disable accounting without affecting the current state of TACACS+ authentication.

- To enable TACACS+ accounting, use the following command:
  `enable tacacs-accounting`
- To disable TACACS+ accounting, use the following command:
  `disable tacacs-accounting`

TACACS+ Accounting Configuration Example

This section provides a sample TACACS+ client configuration for TACACS+ accounting that:

- Specifies the primary TACACS+ accounting server.
- Specifies the shared secret for the primary TACACS+ accounting server.
- Specifies the secondary TACACS+ accounting server.
- Specifies the shared secret for the secondary TACACS+ accounting server.
- Enables TACACS+ accounting on the switch.
- Configures an IPv6 server as the primary TACACS+ accounting server.

All other client configuration features use the default settings as described Configuring the TACACS+ Client for Authentication and Authorization on page 1591 or in the Switch Engine v33.6.1 Command References .

```
configure tacacs-accounting primary server 10.201.31.238 client-ip 10.201.31.85 vr "VR-
Default"
configure tacacs-accounting primary shared-secret purple
configure tacacs-accounting secondary server 10.201.31.235 client-ip 10.201.31.85 vr "VR-
Default"
config tacacs-accounting secondary shared-secret purple
enable tacacs-accounting
```

IPv6 server can be configured using the following commands:

```
configure tacacs-accounting  primary server 2111::220 client-ip 2111::85 vr "VR-Default"
configure tacacs-accounting  primary shared-secret purple
configure tacacs-accounting secondary server 2111::210 client-ip 2111::85 vr "VR-Default"
configure tacacs-accounting secondary shared-secret purple
configure tacacs-accounting primary server 1111::220 client-ip 1111::170 vr vr-mgmt
```

To display the TACACS+ client accounting configuration, use the `show tacacs` or the `show tacacs-accounting` command. The following is sample output from the `show tacacs` command:

```
TACACS+: enabled
TACACS+ Authorization: enabled
TACACS+ Accounting : enabled
TACACS+ Server Connect Timeout sec: 3
Primary TACACS+ Server:
    Server name   :
    IP address    :  10.201.31.238
    Server IP Port:  49
    Client address:  10.201.31.85 (VR-Default)
    Shared secret :  purple
Secondary TACACS+ Server:
    Server name   :
    IP address    :  10.201.31.235
    Server IP Port:  49
```

```
    Client address:  10.201.31.85 (VR-Default)
    Shared secret :  purple
TACACS+ Acct Server Connect Timeout sec: 3
Primary TACACS+ Accounting Server:
    Server name   :
    IP address    :  10.201.31.238
    Server IP Port:  49
    Client address:  10.201.31.85 (VR-Default)
    Shared secret :  purple
Secondary TACACS+ Accounting Server:
    Server name   :
    IP address    :  10.201.31.235
    Server IP Port:  49
    Client address:  10.201.31.85 (VR-Default)
    Shared secret :  purple
```

Authenticating Management Sessions Through a RADIUS Server

You can use a Remote Authentication Dial In User Service (*RADIUS*) server to authenticate management sessions for multiple switches. A RADIUS server allows you to centralize the authentication database, so that you do not have to maintain a separate local database on each switch. RADIUS servers provide the following services for management sessions:

- Username and password authentication
- Command authorization (the RADIUS server validates whether the user is authorized to execute each command)

  Note
  Command usage that should be restricted for a user account by RADIUS with CLI authorization may not occur when users are logged in by Chalet or when using the XML API directly.

- Accounting service (tracks authentication and authorization events)

  Note
  You can use a local database on each switch as a backup authentication service if the RADIUS service is unavailable. When the RADIUS service is operating, privileges defined on the RADIUS server take precedence over privileges configured in the local database.

To use RADIUS server features, you need the following components:

- RADIUS client software, which is included in the ExtremeXOS software.
- A RADIUS server, which is a third-party product.

Note
RADIUS provides many of the same features provided by TACACS+. You cannot use RADIUS and TACACS+ at the same time.

RADIUS is a communications protocol (RFC 2865) that is used between client and server to implement the RADIUS service.

The RADIUS client component of the ExtremeXOS software should be compatible with any RADIUS compliant server product.

Note
The switch allows local authentication when the client IP is excluded in RADIUS server.

The following sections provide more information on management session authentication:

- How Extreme Switches Work with RADIUS Servers on page 1598
- Configuration Overview for Authenticating Management Sessions on page 1599

How Extreme Switches Work with RADIUS Servers

When configured for use with a *RADIUS* server, an ExtremeXOS switch operates as a RADIUS client. In RADIUS server configuration, the client component is configured as a client or as a Network Access Server (NAS). Typically, an ExtremeXOS NAS provides network access to supplicants such as PCs or phones.

When a supplicant requests authentication from a switch that is configured for RADIUS server authentication, the following events occur:

1. The switch sends an authentication request in the form of a RADIUS Access-Request message.
2. The RADIUS server looks up the user in the users file.
3. The RADIUS server accepts or rejects the authentication and returns a RADIUS Access-Accept or Access-Reject message.

   Note
   A user rejected by the Radius/TACACS server can not be authenticated via local database.

4. If authentication is accepted, the Access-Accept message can contain standard RADIUS attributes and Vendor Specific Attributes (VSAs) that can be used to configure the switch.
5. If authentication is accepted, the Access-Accept message can enable command authorization for that user on the switch. Command authorization uses the RADIUS server to approve or deny the execution of each command the user enters.

The ExtremeXOS switch initiates all communications with the RADIUS server. For basic authentication, the switch sends the Access-Request message, and communications with the RADIUS server is complete when the switch receives the Access-Accept or Access-Reject message. For command authorization, communications starts each time a user configured for command authorization enters a switch command. RADIUS server communications ends when command use is allowed or denied.

A key component of RADIUS server management is the attributes and VSAs that the RADIUS server can be configured to send in Access-Accept messages. VSAs are custom attributes for a specific vendor, such as Extreme Networks. These attributes store information about a particular user and the configuration options available to the user. The RADIUS client in ExtremeXOS accepts these attributes and uses them

to configure the switch in response to authentication events. The RADIUS server does not process attributes; it simply sends them when authentication is accepted. It is the switch that processes attributes.

User authentication and attributes are managed on a RADIUS server by editing text files. On the FreeRADIUS server, the user ID, password, attributes, and VSAs are stored in the users file, and VSAs are defined in the dictionary file. The dictionary file associates numbers with each attribute. When you edit the users file, you specify the text version of each attribute you define. When the RADIUS server sends attributes to the switch, it sends the attribute type numbers to reduce the network load. Some attribute values are sent as numbers too.

Command authorization is also managed on a RADIUS server by editing text files. On a FreeRADIUS server, the profiles file is divided into sections called *profiles*. Each profile lists command access definitions. In the users file, you can use the Profile-Name attribute to select the command profile that applies to each user managed by command authorization.

The ExtremeXOS software supports backup authentication and authorization by eight total servers for redundancy.

RADIUS servers can be optionally configured to work with directory services such as LDAP or Microsoft Active Directory. Because ExtremeXOS switches operate with RADIUS servers, they can benefit from the pairing of the RADIUS server and a directory service. Some guidelines for configuring FreeRADIUS with LDAP are provided later in this chapter. Since the use of the directory service requires configuration of the RADIUS server and directory service, the appropriate documentation to follow is the documentation for those products.

Configuration Overview for Authenticating Management Sessions

To configure the switch *RADIUS* client and the RADIUS server to authenticate management sessions, do the following:

1. Configure the switch RADIUS client for authentication as described in Configuring the RADIUS Client for Authentication and Authorization on page 1611.
2. If you want to use RADIUS accounting, configure the switch RADIUS accounting client as described in Configuring the RADIUS Client for Accounting on page 1613.
3. Configure the RADIUS server for authentication as described in Configuring User Authentication (Users File) on page 1615.
4. If you want to use RADIUS accounting, configure a RADIUS accounting server as described in the documentation for your RADIUS product.

Authenticating Network Login Users Through a RADIUS Server

You can use a *RADIUS* server to authenticate network login users and supply configuration data that the switch can use to make dynamic configuration changes to accommodate network login users.

A RADIUS server allows you to centralize the authentication database, so that you do not have to maintain a separate local database on each switch. RADIUS servers provide the following services for network login sessions:

- Username and password authentication
- Standard RADIUS attributes and Extreme Networks VSAs that the switch can use for dynamic configuration
- Accounting service (tracks authentication and authorization events)

To use RADIUS server features, you need the following components:

- RADIUS client software, which is included in the ExtremeXOS software.
- A RADIUS server, which is a third-party product.

Note
RADIUS provides many of the same features provided by TACACS+, but the network login feature does not work with TACACS+.

The following sections provide more information on network login session authentication:

- Differences Between Network Login Authentication and Management Session Authentication on page 1600
- Configuration Overview for Authenticating Network Login Users on page 1600

Differences Between Network Login Authentication and Management Session Authentication

Network login authentication is very similar to management session authentication.

The differences are:

- Network login authentication grants network access to devices connected to a switch port, and management session authentication grants management access to the switch for configuration and management.
- The user name for network login authentication can be a MAC address.
- Standard *RADIUS* attributes and Extreme Networks VSAs can be used with the network login and universal port features to configure switch ports and general switch configuration parameters.
- Command authorization is not applicable because network login controls network access, not management session access.

Except for the above differences, network login authentication is the same as described in How Extreme Switches Work with RADIUS Servers on page 1598.

Configuration Overview for Authenticating Network Login Users

To configure the switch *RADIUS* client and the RADIUS server to authenticate network login users, do the following:

1. Configure the switch RADIUS client for authentication as described in Configuring the RADIUS Client for Authentication and Authorization on page 1611.
2. If you want to use RADIUS accounting, configure the switch RADIUS accounting client as described in Configuring the RADIUS Client for Accounting on page 1613.

3. Configure network login on the switch as described in Security.
4. Configure the RADIUS server for authentication and Extreme Networks VSAs as described in Configuring User Authentication (Users File) on page 1615.
5. If you want to use the universal port feature to run configuration scripts at authentication, configure the switch universal port feature as described in Universal Port on page 2135.
6. If you want to use RADIUS accounting, configure a RADIUS accounting server as described in the documentation for your RADIUS product.

Authentication

Entire packets (as opposed to individual TLVs in a packet) can be authenticated. If authentication fails on a packet, the entire packet is discarded. Multi-part packets are authenticated separately. Routers are not required to support authentication. The ExtremeXOS software provides optional support for plain-text authentication.

Authentication Retransmission Algorithm

Two retransmission algorithms are used in combination: Back-off Round Robin, and simple Round Robin. The focus of this retransmission algorithm is to provide for server redundancy.

Note
The reason for using a combination of back-off and round-robin rather than the standard back-off algorithm where all configured transmissions occur to server 1 before transmitting to server 2 is to allow for more than one server to be tried prior to 802.1x timeout when EAP authentication is occurring.

The standard algorithm is as described in Figure 195 and is a combination of back-off and round-robin. This algorithm always uses the highest priority server first regardless of past transaction history. If the highest priority server can handle the entire load all, transactions will go to that server. Consider three *RADIUS* servers – 1, 2 and 3 with the configurable number of retries set to 2:

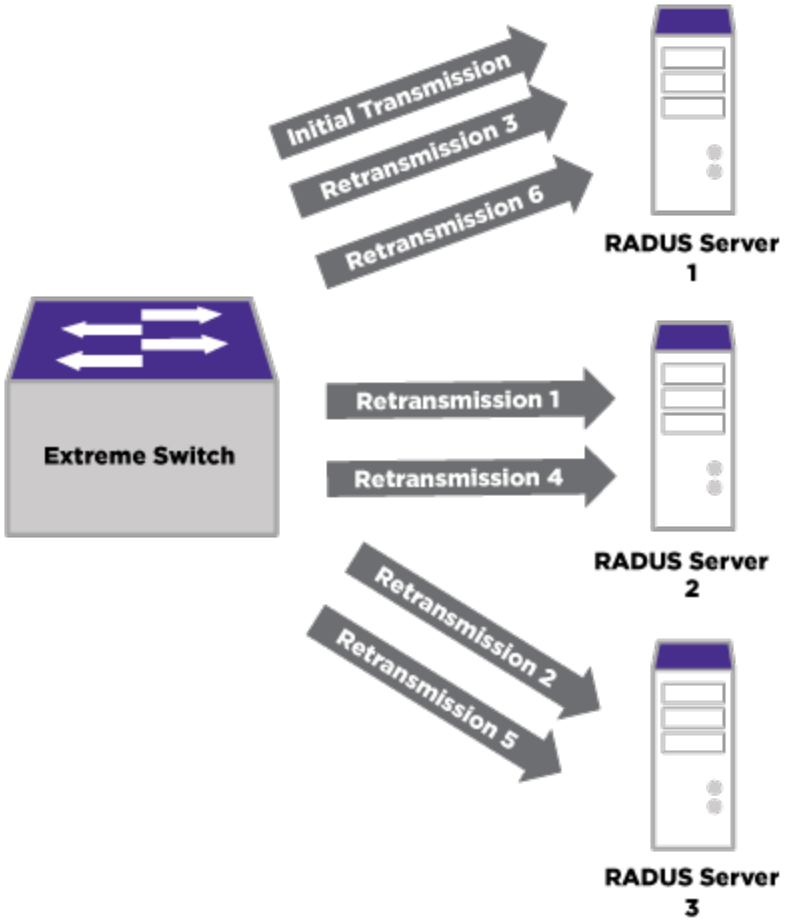


Figure 195: Authentication Retransmission Algorithm for a Single RADIUS Transaction (No Servers Responding 1)

This figure shows the entire retransmission algorithm for a single RADIUS transaction if none of the servers were to respond. No more transmissions will occur for this transaction if a response is received by the RADIUS client software within the configurable timeout period.

The round-robin retransmission algorithm is depicted in Figure 196 and is simply round-robin.

The configurable round-robin retransmission algorithm for RADIUS authentication aims to spread the load among all the configured servers. In large-scale deployments with high rates of authentication this algorithm will provide for better performance than the default algorithm. The initial transmission for each potential authentication will go to the next server in the list. If 999 sessions were to be authenticated across three servers and no timeouts were to occur, then 333 responses would be sent to each server.

Consider three RADIUS servers: 1, 2 and 3 with the configurable number of retries set to 2 and where the prior session sent its initial request to server 1:

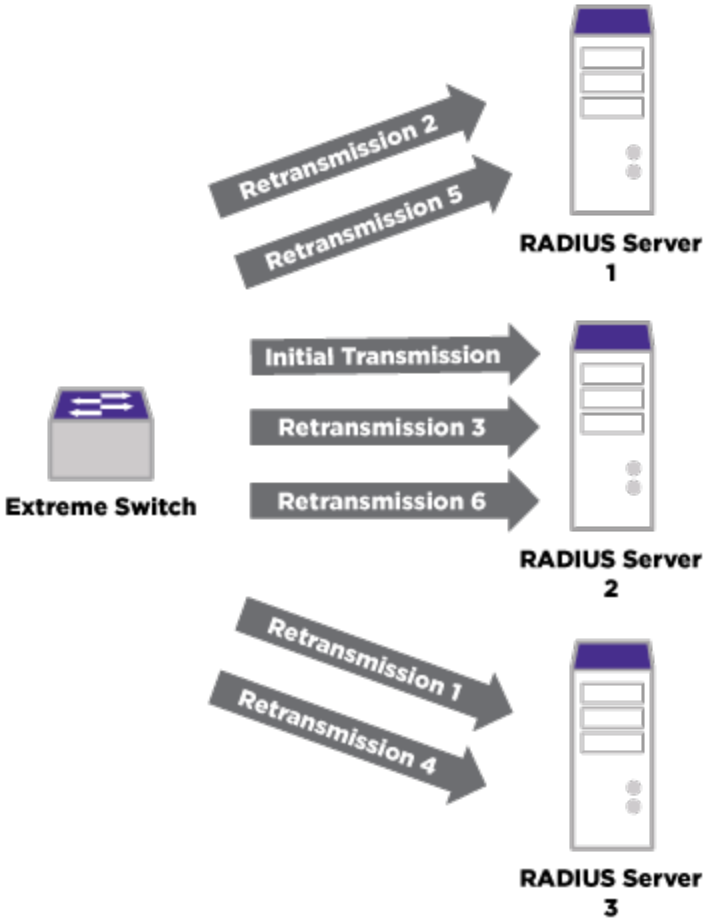


Figure 196: Authentication Retransmission Algorithm for a Single RADIUS Transaction (No Servers Responding 2)

This figure shows the entire retransmission algorithm for a single RADIUS transaction if none of the servers were to respond. No more transmissions will occur for this transaction if a response is received by the RADIUS client software within the configurable timeout period. All servers are considered the same priority when using this transmission algorithm with each server taking its turn receiving the initial transmission.

In version 33.4.1, the RADIUS authentication mechanism is enhanced to improve reliability in 802.1X EAP workflows. For dot1x EAP authentication, retransmissions of Access-Request packets will now be attempted on the same RADIUS server that issued the challenge.

Configurable Web Authentication Token Time-to-Live

Version 33.6.1 adds the ability to configure the time-to-live (TTL) for web authentication tokens. This enables shorter token lifetimes for automated tools and enhanced security while maintaining flexibility for different use cases.

Token Authentication Overview

The web interface supports two authentication mechanisms:

- **Basic authentication:** Username and password passed with each request
- **Token authentication:** Token generated once and reused for multiple requests during its validity period

Token authentication is particularly useful for automation tools and scripts that make multiple API calls, as it reduces the overhead of authenticating each request. Previously, authentication tokens were issued with a fixed 24-hour (86400 seconds) validity period. You can now configure a custom default TTL between 1 minute and 24 hours.

Configuring Token TTL

To configure the default token time-to-live:

```
configure web authentication token default-ttl <seconds>
```

Where *seconds* specifies the validity time in seconds (60-86400, default 86400).

Examples:

```
# Set 1-minute TTL for short-lived automation tasks
configure web authentication token default-ttl 60

# Set 1-hour TTL
configure web authentication token default-ttl 3600

# Restore default 24-hour TTL
configure web authentication token default-ttl 86400
```

The configuration persists across reboots.

Default TTL Behavior

The configured default TTL applies when an API request for token generation does not include a TTL property in the request body. If an API request specifies a TTL value, that value takes precedence over the configured default.

This allows individual API clients to request specific TTL values while providing a system-wide default for clients that do not specify a TTL.

Viewing Token TTL Configuration

To view the configured token TTL in the system configuration:

```
show configuration thttpd
```

Sample output:

```
#
# Module thttpd configuration.
#
enable web http
enable web https
configure ssl certificate hash-algorithm sha512
configure web authentication token default-ttl 60
```

To view the token TTL in switch management information:

```
show switch management
```

Sample output excerpt:

```
Web access                          : Enabled (tcp port 80)
                                    : Access Profile : not set
                                    : Auth token default TTL: 60 seconds
```

Generating Authentication Tokens

Authentication tokens are generated via API calls to the `/auth/token` endpoint. The token request includes a username and password:

```
curl --request POST \
  --url http://<switch-ip>/auth/token \
  --header 'content-type: application/json' \
  --data '{
  "username": "admin",
  "password": ""
}'
```

Sample response showing the configured TTL:

```
{
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...",
  "ttl": 60
}
```

The `ttl` field in the response indicates the token's validity period in seconds.

Using Authentication Tokens

Include the authentication token in subsequent API requests using the `x-auth-token` header:

```
curl --request GET \
  --url 'http://<switch-ip>/rest/openapi/v0/state/slpp' \
  --header 'x-auth-token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...'
```

The token remains valid for the configured TTL period from the time it was generated. After the TTL expires, API requests using that token will fail with an authentication error, and a new token must be obtained.

Use Cases and Best Practices

Short-Lived Tokens (60-3600 seconds):

Appropriate for automation tools like Ansible modules that execute brief tasks. Shorter token lifetimes reduce the risk window if a token is compromised. Use short-lived tokens when:

- Running automated configuration scripts with a defined duration
- Integrating with CI/CD pipelines where jobs complete quickly
- Executing one-time API operations
- Security policies require frequent re-authentication

Standard Tokens (3600-86400 seconds):

Suitable for interactive sessions or longer-running automation tasks requiring extended token validity. Use standard tokens when:

- Running long-duration monitoring or data collection scripts
- Providing API access for interactive development and testing
- Integrating with management systems that maintain persistent connections
- Balancing security with operational convenience

Security Considerations:

- Shorter token lifetimes enhance security by limiting the time window during which a compromised token can be used
- Balance security requirements against operational needs when selecting a TTL value
- Store tokens securely in your automation tools and scripts
- Implement token refresh logic in long-running applications to handle token expiration gracefully
- Use HTTPS for all API communication to protect tokens in transit

Operational Considerations:

- Very short TTLs (60-300 seconds) may require frequent token regeneration in automation scripts
- Ensure your automation tools handle token expiration and renewal appropriately
- Monitor authentication failures that may indicate token expiration issues
- Document the configured TTL value for administrators and automation developers

Integration with Automation Tools

When integrating with automation tools such as Ansible:

1. Configure an appropriate token TTL based on typical task duration
2. Generate an authentication token at the start of the automation workflow
3. Use the token for all API requests during the workflow
4. Implement error handling to detect token expiration and generate a new token if needed
5. Store tokens securely using your automation platform's secret management features

Example workflow for a 5-minute Ansible playbook:

```
# Configure 10-minute token TTL to allow buffer time
configure web authentication token default-ttl 600

# Ansible playbook uses token authentication
# Token obtained at playbook start
# Token valid for entire playbook execution
# Token automatically expires after 10 minutes
```

Accounting

Accounting Start and Stop requests are sent for user sessions in accordance with RFC 2866 (7).

Accounting Retransmission Algorithm

The *RADIUS* client accounting software uses the standard back-off retransmission algorithm. Since accounting transactions generally do not require timeliness this algorithm focuses on redundancy rather than expediency. The highest priority server is always tried first regardless of past transaction history. Consider three accounting servers – 1, 2 and 3 each with the configurable number of retries set to 2:

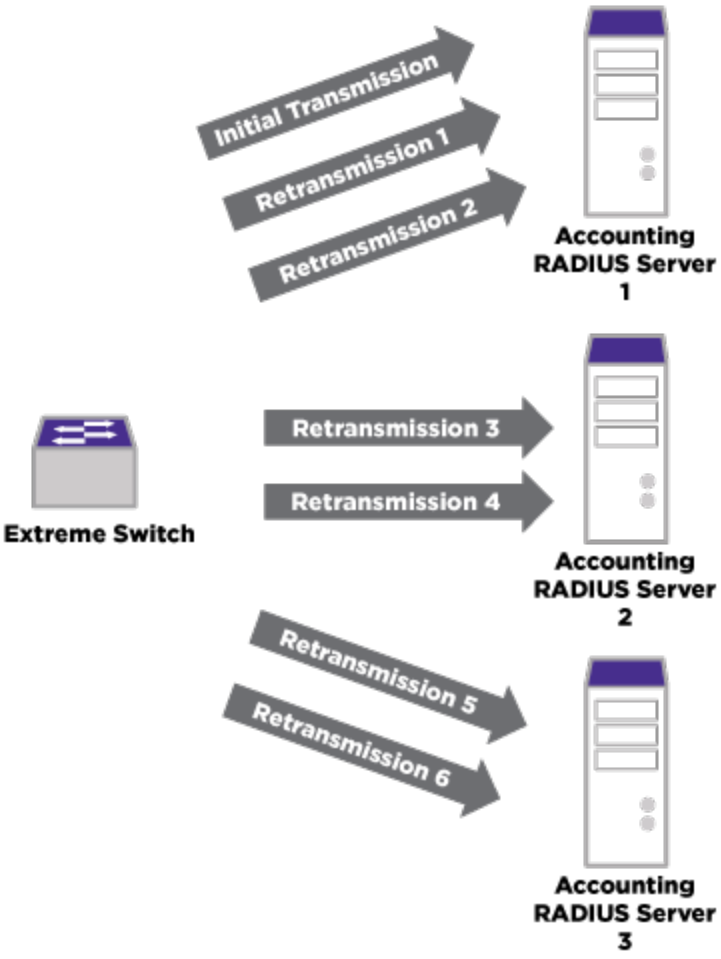


Figure 197: Accounting Retransmission Algorithm (No Servers Responding)

This figure shows the entire retransmission algorithm for a single RADIUS accounting transaction if none of the servers were to respond. All of the transactions to each of the servers are attempted before another server is attempted. If the initial server can handle the entire accounting transmission load, then all transactions will go to that server.

Authentication NMS Realm

The NMS realm type is supported in addition to the management and network realms. These servers are used for the Extreme Network Virtualization or Virtual Machine Tracking feature. This realm only supports the existing subset of two servers (primary and secondary).

Per Realm Authentication Enable/Disable

RADIUS authentication can be enabled on a per realm basis for the management and network realms.

Supported RADIUS Attributes

The following is a list of *RADIUS* attributes.

Note
Although this is the list of attributes that is supported by the RADIUS software, usage of these attributes is dependent on the features provided by the system as a whole.

Table Legend:

- PRQ—PAP Authentication Request
- CRQ—CHAP Authentication Request
- EARQ—EAP Authentication Request
- AC—Access Challenge
- AA—Access Accept
- AR—Access Reject

Table 143: Authentication Attributes

Attribute Name	RFC	Attr #	PRQ	CRQ	EARQ	AC	AA	AR
User-Name	2865	1	X	X	X		X	X
User-Password	2865	2	X					
CHAP-Password	2865	3		X				
NAS-IP-Address	2865	4	X	X	X			
NAS-Port	2865	5	X	X	X			
Service-Type	2865	6	X	X	X		X	
Framed-Protocol	2865	7		X	X			
Filter-ID	2865	11					X	
Framed-MTU	2865	12	X	X	X			
State	2865	24			X	X		
Class	2865	25					X	
Session-Timeout	2865	27					X	

Table 143: Authentication Attributes (continued)

Attribute Name	RFC	Attr #	PRQ	CRQ	EARQ	AC	AA	AR
Idle-Timeout	2865	28					X	
Termination-Action	2865	29					X	
Called-Station-ID	2865	30	X	X	X			
Calling-Station-ID	2865	31	X	X	X			
NAS-Identifier	2865	32	X	X	X			
CHAP-Challenge	2865	60		X				
NAS-Port-Type	2865	61	X	X	X			
Tunnel-Type	2868	64					X	
Tunnel-Medium	2868	65					X	
Message-Authenticator	2869	80			X	X	X	X
Tunnel-Private-Group-ID	2868	81					X	
Acct-Interim-Interval	2868	85					X	
NAS-IPV6-Address	3162	95	X	X	X			
MS-MPPE-Send-Key	2548	16					X	
MS-MPPE-Recv-Key	2548	17					X	
MS-Quarantine-State	VSA	45					X	X
MS-IPv4-Remediation-servers	VSA	52					X	X
FA-VLAN-ISID VSA 171	VSA	171					X	
Fabric-Attach-Service-Request	VSA	186					X	
Extreme-CLI-Authorization	VSA	201					X	
Extreme-Shell-Command	VSA	202	X					
Extreme-Netlogin-*VLAN*	VSA	203					X	
Extreme-Netlogin-URL	VSA	204					X	
Extreme-Netlogin-URL-Desc	VSA	205					X	
Extreme-Netlogin-Only	VSA	206					X	
Extreme-Netlogin-VLAN-Tag	VSA	209					X	

Table 143: Authentication Attributes (continued)

Attribute Name	RFC	Attr #	PRQ	CRQ	EARQ	AC	AA	AR
Extreme-Netlogin-Extended-VLAN	VSA	211					X	
Extreme-Security-Profile	VSA	212					X	
Extreme-VM-Name	VSA	213					X	
Extreme-VM-VPP-Name	VSA	214					X	
Extreme-VM-IP-Addr	VSA	215					X	
Extreme-VM-VLAN-Tag	VSA	216					X	
Extreme-VM-VR-Name	VSA	217					X	
Extreme-NetLogin-VR-Name	VSA	218					X	
Extreme-Policy-ACL	VSA	232					X	
Extreme-Port-Bounce	VSA	233					X	
Extreme-URL-Redirect VSA	VSA	234					X	

Table Legend:

- ASTA—Accounting Start Request
- ASTO—Accounting Stop Request
- AR—Accounting Response

Table 144: Accounting Attributes

Attribute Name	RFC	Attr #	ASTA	ASTO	AR
User-Name	2865	1	X	X	
NAS-IP-Address	2865	4	X	X	
NAS-Port	2865	4	X	X	
Class	2865	25	X	X	
Calling-Station-ID	2865	31	X	X	
Acct-Status-Type	2866	40	X	X	
Acct-Delay-Time	2866	41	X	X	
Acct-Session-ID	2866	44	X	X	
Acct-Authentic	2866	45	X	X	
Acct-Session-Time	2866	46		X	
Acct-Terminate-Cause	2866	49		X	

Table 144: Accounting Attributes (continued)

Attribute Name	RFC	Attr #	ASTA	ASTO	AR
NAS-IPV6-Address	3162	95	X	X	
ETS-Auth-Client-Type	VSA	1	X	X	
Acct-Input-Octets	2866	42		X	
Acct-Output-Octets	2866	43		X	
Acct-Input-Packets	2866	47		X	
Acct-Output-Packets	2866	48		X	
Login-Service	2865	15		X	

Configuring the RADIUS Client

For information on installing, configuring, and managing a *RADIUS* server, see the product documentation for that server and the guidelines in RADIUS Server Configuration Guidelines on page 1615.

Configuring the RADIUS Client for Authentication and Authorization

Specifying the RADIUS Server Addresses

Before the *RADIUS* client software can communicate with a RADIUS server, you must specify the server address in the client software. You can specify up to 32 RADIUS servers, and you can use either an IP address or a host name to identify each server.

- To configure the RADIUS servers in the client software, use the following command:

  `configure radius {mgmt-access | netlogin} [primary | secondary | index] server [host_ipaddr | host_ipV6addr | hostname] {udp_port} client-ip [client_ipaddr | client_ipV6addr] {vr vr_name} {shared-secret {encrypted} secret}`

  Note
  It is recommended to enable loopback mode on the *VLAN* associated with RADIUS if the radius connectivity is established via a front panel port on a SummitStack.

  The default port value for authentication is 1812. The client IP address is the IP address used by the RADIUS server for communicating back to the switch.

  To configure the primary RADIUS server, specify primary. To configure the secondary RADIUS server, specify secondary.

  By default, switch management and network login use the same primary and secondary RADIUS servers for authentication. To specify one pair of RADIUS servers for switch management and another pair for network login, use the mgmt-access and netlogin keywords.

Configuring the RADIUS Client Timeout Value

- To configure the timeout if a server fails to respond, use the following command:
  `configure radius {mgmt-access {primary | secondary} | netlogin {primary | secondary} | index } timeout sec`

  If the timeout expires, another authentication attempt is made. After three failed attempts to authenticate, the alternate server is used. After six failed attempts, local user authentication is used. If the user does not have a local account or the user is disabled locally, the user's login will fail.

  If you do not specify the mgmt-access or netlogin keyword, the timeout interval applies to both switch management and netlogin *RADIUS* servers.

Configuring the Shared Secret Password for RADIUS Communications

The shared secret is a password that is configured on each network device (*RADIUS* client) and RADIUS server. The shared secret is used to verify communication between network devices and the server.

- To configure the shared secret for client communications with RADIUS servers, use the following command:
  `configure radius {mgmt-access | netlogin} {primary | secondary} shared-secret {encrypted} string`

  To configure the shared secret for a primary RADIUS server, specify primary. To configure the shared secret for a secondary RADIUS server, specify secondary.

  If you do not specify the mgmt-access or netlogin keyword, the secret applies to both the primary and secondary switch management and network login RADIUS servers.

  Do not use the encrypted keyword to set the shared secret. The encrypted keyword prevents the display of the shared secret in the `show configuration` command output.

Enabling and Disabling the RADIUS Client Service

The *RADIUS* client service can be enabled or disabled without affecting the client configuration. When the client service is disabled, the client does not communicate with the RADIUS server, so authentication must take place through the another service such as the local database or a TACACS+ server.

Note
You cannot use RADIUS and TACACS+ at the same time.

- To enable the RADIUS client service, use the following command:
  `enable radius {mgmt-access | netlogin}`
- To disable the RADIUS client service, use the following command:
  `disable radius {mgmt-access | netlogin}`

  If you do not specify the mgmt-access or netlogin keywords, RADIUS authentication is enabled or disabled on the switch for both management and network login.

Configuring the RADIUS Client for Accounting

Specifying the RADIUS Accounting Server Addresses

Before the *RADIUS* client software can communicate with a RADIUS accounting server, you must specify the server address in the client software. You can specify up to 32 accounting servers, and you can use either an IP address or a host name to identify each server.

- To specify RADIUS accounting servers, use the following command:

  `configure radius-accounting { mgmt-access | netlogin } [ primary | secondary | index ] server [ host_ipaddr | host_ipV6addr | hostname] {udp_port} client-ip [ client_ipaddr | client_ipV6addr] {vr vr_name} {shared-secret {encrypted} secret}`

  The default port value for accounting is 1813. The client IP address is the IP address used by the RADIUS server for communicating back to the switch.

  To configure the primary RADIUS accounting server, specify primary. To configure the secondary RADIUS accounting server, specify secondary.

  By default, switch management and network login use the same primary and secondary RADIUS servers for accounting. To specify one pair of RADIUS accounting servers for switch management and another pair for network login, use the mgmt-access and netlogin keywords.

Configuring the RADIUS Client Accounting Timeout Value

- To configure the timeout if a server fails to respond, use the following command:

  `configure radius-accounting {mgmt-access {primary | secondary} | netlogin {primary | secondary} | index } timeout sec`

  If the timeout expires, another authentication attempt is made. After three failed attempts to authenticate, the alternate server is used.

Configure the Shared Secret Password for RADIUS Accounting Servers

The shared secret is a password that is configured on each network device (*RADIUS* client) and RADIUS accounting server. The shared secret is used to verify communication between network devices and the server.

- To configure the shared secret for client communications with RADIUS accounting servers, use the following command:

  `configure radius-accounting {mgmt-access | netlogin} [primary | secondary] shared-secret {encrypted} string`

  To configure the primary RADIUS accounting server, specify primary. To configure the secondary RADIUS accounting server, specify secondary.

  If you do not specify the mgmt-access or netlogin keywords, the secret applies to both the primary and secondary switch management and network login RADIUS accounting servers.

Do not use the encrypted keyword to set the shared secret. The encrypted keyword prevents the display of the shared secret in the `show configuration` command output.

Enabling and Disabling RADIUS Accounting

After you configure the *RADIUS* client with the RADIUS accounting server information, you must enable accounting in the RADIUS client before the switch begins transmitting the information. You must enable RADIUS authentication in the client for accounting information to be generated. You can enable and disable accounting without affecting the current state of RADIUS authentication.

- To enable RADIUS accounting, use the following command:

  `enable radius-accounting {mgmt-access | netlogin}`

- To disable RADIUS accounting, use the following command:

  `disable radius-accounting {mgmt-access | netlogin}`

  If you do not specify a keyword, RADIUS accounting is enabled or disabled on the switch for both management and network login.

VLAN Name in RADIUS and TACACS Configurations

Version 33.4.1 introduces support for using VLAN names in place of static IP addresses when configuring RADIUS and TACACS servers. This enhancement allows the switch to dynamically resolve the IP address associated with a specified VLAN, reducing the need for manual updates when IP addresses change.

What this feature does:

- Simplifies configuration by referencing VLAN names instead of hardcoded IPs.
- Automatically uses the primary IPv4 or lowest IPv6 address from the VLAN.
- Enhances flexibility and reduces administrative overhead.

Configuration examples:

```
configure radius primary server <server_ip> client-vlan <vlan_name> vr <vr_name> shared-
secret <secret>
configure tacacs primary server <server_ip> client-vlan <vlan_name> vr <vr_name>
```

Behavior Notes and Limitations

- If the VLAN is not yet created, the configuration is accepted but marked as "Not In Service" until the VLAN and its IP are available.
- If a VLAN has multiple IPs:
  - For IPv4: the primary address is used.
  - For IPv6: the lowest address is used.
- Dynamic VLAN creation (for example, using ZTP+ or MVRP) is supported.

Limitations:

SNMP MIB support for client-vlan is not included.

RADIUS Server Configuration Guidelines

Note
For information on how to use and configure your *RADIUS* server, refer to the documentation that came with your RADIUS server.

Note
When configuring RADIUS/TACACS servers with the IP address received from a DHCP server, the IP address should be reserved for the switch in the DHCP server to avoid conflicts after rebooting.

Configuring User Authentication (Users File)

User authentication is configured in the users file on a FreeRADIUS server. Other *RADIUS* servers might use a different name and a different syntax for configuration, but the basic components of the users file and user authentication are the same.

For Extreme Networks switches, there are three types of users file entries:

- Session management entries
- Network login user entries
- Network login MAC address entries

Note
The “users” file is case-sensitive, and punctuation is very important for FreeRADIUS.

Session Management Entries

The following is an example of a session management entry for read-write access:

```
eric
Cleartext-Password := "eric"
Service-Type = Administrative-User,
Extreme-CLI-Authorization = Enabled
```

The key components of the example above are the user name, password, service-type, and Extreme-CLI-Authorization VSA. For simple authentication, you only need to enter the user name ("eric" in this example) and a password as described in the *RADIUS* server documentation.

Enter the attributes for each user and separate them from the others with commas as described in the RADIUS server documentation.

For more information on the Extreme-CLI-Authorization VSA, see Extreme Networks VSAs on page 1619.

Network Login User Entries

The following is an example of a network login user entry:

```
Jim             Auth-Type := EAP, Cleartext-Password == "12345"
                Session-Timeout = 60,
```

```
                Termination-Action = 1,
                Extreme-Security-Profile = "user-auth LOGOFF-PROFILE=avaya-
                  remove;qos=\"QP1\";",
                Extreme-Netlogin-Vlan = voice-avaya
```

The key components of the example above are the user name, password, attributes, and Extreme Networks VSAs. For simple authentication, you only need to enter the user name (Jim in this example) and a password as described in the *RADIUS* server documentation.

Enter the attributes for each user and separate them from the others with commas as described in the RADIUS server documentation.

In the example above, the Session-Timeout and Termination-Action attributes are examples of standard RADIUS attributes, and these are described in Standard RADIUS Attributes Used by Extreme Switches on page 1616.

The Extreme-Security-Profile and Extreme-Netlogin-Vlan attributes are examples of Extreme Networks VSAs and are described in Extreme Networks VSAs on page 1619.

Network Login MAC Address Entries

The following is an example of a network login MAC address entry:

```
00040D9D12AF
Auth-Type := Local,
Cleartext-Password == "00040D9D12AF"
Session-Timeout = 60,
Termination-Action = 1,
Extreme-Security-Profile = "user-auth LOGOFF-PROFILE=avaya remove;qos=\"QP1\";",
Extreme-Netlogin-Vlan = voice-avaya
```

The key components of the example above are the MAC address, password (which is set to the MAC address), attributes, and Extreme Networks VSAs. For simple authentication, you only need to enter the MAC address (00040D9D12AF in this example) and a password as described in the *RADIUS* server documentation.

Enter the attributes for each user and separate them from the others with commas as described in the RADIUS server documentation.

In the example above, the Session-Timeout and Termination-Action attributes are examples of standard RADIUS attributes, and these are described in Standard RADIUS Attributes Used by Extreme Switches on page 1616.

The Extreme-Security-Profile and Extreme-Netlogin-*VLAN* attributes are examples of Extreme Networks VSAs and are described in Extreme Networks VSAs on page 1619.

Standard RADIUS Attributes Used by Extreme Switches

The ExtremeXOS software uses standard *RADIUS* attributes to send information in an Access-Request message to a RADIUS server.

The software also accepts some standard RADIUS attributes in the Access-Accept message that the RADIUS server sends to the switch after successful authentication. The switch ignores attributes that it is not programmed to use.

Table 145 lists the standard RADIUS attributes used by the ExtremeXOS software.

Table 145: Standard RADIUS Attributes Used by Network Login

Attribute	RFC	Attribute Type	Format	Sent-in	Description
User-Name	RFC 2138	1	String	Access-Request	Specifies a user name for authentication.
Calling-Station-ID	RFC 2865	31	String	Access-Request	Identifies the phone number for the supplicant requesting authentication.
EAP-Message	RFC 3579	79	String	Access-Request, Access-Challenge, Access-Accept, and Access Reject	Encapsulates EAP packets.
Login-IP-Host	RFC 2138	14	Address	Access-Request and Access-Accept	Specifies a host to log into after successful authentication.
Message-Authenticator	RFC 3579	80	String	Access-Request, Access-Challenge, Access-Accept, and Access Reject	Contains a hash of the entire message that is used to authenticate the message.
NAS-Port-Type	RFC 2865	61	Integer	Access-Request	Identifies the port type for the port through which authentication is requested.
Service-Type	RFC 2138	6	String	Access-Accept	Specifies the granted service type in an Access-Accept message. See Attribute 6: Service Type below.
Session-Timeout	RFC 2865	27	Integer	Access-Accept, Access-Challenge	Specifies how long the user session can last before authentication is required.
State	RFC 2865	24	String	Access-Challenge, Access-Request	Site specific.
Termination-Action	RFC 2865	29	Integer	Access-Accept	Specifies how the switch should respond to service termination.

Table 145: Standard RADIUS Attributes Used by Network Login (continued)

Attribute	RFC	Attribute Type	Format	Sent-in	Description
Tunnel-Medium-Type	RFC 2868	65	Integer	Access-Accept	Specifies the transport medium used when creating a tunnel for protocols (for example, VLANs) that can operate over multiple transports.
Tunnel-Private-Group-ID	RFC 2868	81	Integer/String	Access-Accept	Specifies the *VLAN* ID of the destination VLAN after successful authentication; used to derive the VLAN name. Usually, the Integer VLAN ID (1-4094) is used. However, If the desired VLAN is statically configured and bound to a specific VLAN ID, the VLAN name string can be used for this attribute as well.
Tunnel-Type	RFC 2868	64	Integer	Access-Accept	Specifies the tunneling protocol that is used.
User-Password	RFC 2138	2	String	Access-Request	Specifies a password for authentication.

Attribute 6: Service Type

During authentication, ExtremeXOS, in conformance with *RFC 2865*, sends the following values for the following authentication methods:

- Web-Auth—“Login"
- MAC-Auth—“Call Check”
- Dot1x—“Framed”

Extreme Networks switches have two levels of user privilege:

- read-only
- read-write

Because no command line interface (CLI) commands are available to modify the privilege level, access rights are determined when you log in. For a RADIUS server to identify the administrative privileges of a user, Extreme Networks switches expect a RADIUS server to transmit the Service-Type attribute in the Access-Accept packet, after successfully authenticating the user.

Extreme Networks switches grant a RADIUS-authenticated user read-write privilege if a Service-Type value of 6 is transmitted as part of the Access-Accept message from the RADIUS server. Other Service-Type values or no value, result in the switch

granting read-only access to the user. Different implementations of RADIUS handle attribute transmission differently. You should consult the documentation for your specific implementation of RADIUS when you configure users for read-write access.

Note
When a RADIUS Access-Accept message is received with a Termination-Action attribute value of RADIUS-Request and a Session-Timeout attribute value of zero, reauthentication occurs immediately, as specified in Section 3.17 of RFC 3580. These RADIUS server configuration settings can cause continuous reauthentication of 802.1X clients.

Extreme Networks VSAs

Table 146 contains the Vendor Specific Attribute (VSA) definitions that a *RADIUS* server can send to an Extreme switch after successful authentication.

These attributes must be configured on the RADIUS server along with the Extreme Networks Vendor ID, which is 1916.

Table 146: VSA Definitions for Web-Based, MAC-Based, and 802.1X Network Login

VSA	Attribute Type	Format	Sent-in	Description
Extreme-CLI-Authorization	201	Integer	Access-Accept	Specifies whether command authorization is to be enabled or disabled for the user on the ExtremeXOS switch.
Extreme-Netlogin-VLAN-Name	203	String	Access-Accept	Name of destination VLAN after successful authentication (must already exist on switch).
Extreme-Netlogin-URL	204	String	Access-Accept	Destination web page after successful authentication.
Extreme-Netlogin-URL-Desc	205	String	Access-Accept	Text description of network login URL attribute.
Extreme-Netlogin-Only	206	Integer	Access-Accept	Indication of whether the user can authenticate using other means, such as telnet, console, SSH, or Vista. A value of “1” (enabled) indicates that the user can only authenticate via network login. A value of “0” (disabled) indicates that the user can also authenticate via other methods.
Extreme-User-Location	208	String		
Extreme-Netlogin-VLAN-ID	209	Integer	Access-Accept	ID of destination VLAN after successful authentication (except for dynamic VLANs, must already exist on switch).

Table 146: VSA Definitions for Web-Based, MAC-Based, and 802.1X Network Login (continued)

VSA	Attribute Type	Format	Sent-in	Description
Extreme-Netlogin-Extended-VLAN	211	String	Access-Accept	Name or ID of the destination VLAN after successful authentication (must already exist on switch). **Note:** When using this attribute, specify whether the port should be moved tagged or untagged to the VLAN. See the guidelines listed in the section VSA 211: Extreme-Netlogin-Extended-Vlan on page 1623 for more information.
Extreme-Security-Profile	212	String	Access-Accept	Specifies a universal port profile to execute on the switch. For more information, see Universal Port on page 2135.
EXTREME_VM_NAME	213	String	Access-Accept	Specifies the name of the VM that is being authenticated . Example: MyVM1
EXTREME_VM_VPP_NAME	214	String	Access-Accept	Specifies the VPP to which the VM is to be mapped. Example: nvpp1
EXTREME_VM_IP_ADDR	215	String	Access-Accept	Specifies the IP address of the VM . Example: 11.1.1.254
EXTREME_VM_CTag	216	Integer	Access-Accept	Specifies the ID or tag of the destination VLAN for the VM . Example: 101
EXTREME_VM_VR_Name	217	String	Access-Accept	Specifies the VR in which the destination VLAN is to be placed. Example : UserVR1

The examples in the following sections are formatted for use in the FreeRADIUS users file. If you use another RADIUS server, the format might be different.

Note

For information on how to use and configure your RADIUS server, refer to the documentation that came with your RADIUS server.

For untagged VLAN movement with 802.1X netlogin, you can use all current Extreme Networks VLAN VSAs: VSA 203, VSA 209, and VSA 211.

VSA 201: Extreme-CLI-Authorization

This attribute specifies whether command authorization is to be enabled or disabled for the user on the ExtremeXOS switch.

If command authorization is disabled, the user has full access to all CLI commands. If command authorization is enabled, each command the user enters is accepted or rejected based on the contents of the profiles file on the RADIUS server.

When added to the RADIUS users file, the following example enables command authorization for the associated user:

Extreme: Extreme-CLI-Authorization = enabled

When added to the RADIUS users file, the following example disables command authorization for the associated user:

```
Extreme: Extreme-CLI-Authorization = disabled
```

VSA 203: Extreme-Netlogin-VLAN-Name

This attribute specifies a destination VLAN name that the RADIUS server sends to the switch after successful authentication.

The VLAN must already exist on the switch. When the switch receives the VSA, it adds the authenticated user to the VLAN.

The following describes the guidelines for VSA 203:

- For untagged VLAN movement with 802.1X netlogin, you can use all current Extreme Networks VLAN VSAs: VSA 203, VSA 209, and VSA 211.
- To specify the VLAN name, use an ASCII string.
- When using this VSA, do not specify whether the VLAN is tagged or untagged.

Because the RADIUS server can identify a target VLAN with multiple attributes, the switch selects the appropriate VLAN or VLANs using the order:

- Extreme-Netlogin-Extended-VLAN (VSA 211)
- Extreme-Netlogin-VLAN-Name (VSA 203)
- Extreme-Netlogin-VLAN-ID (VSA 209)
- Tunnel-Private-Group-ID, but only if Tunnel-Type == VLAN(13) and Tunnel-Medium-Type == 802 (6) (see Standard RADIUS Attributes Used by Extreme Switches on page 1616)

If none of the previously described attributes are present ISP mode is assumed, and the client remains in the configured VLAN.

When added to the RADIUS users file, the following example specifies the destination VLAN name, purple, for the associated user:

```
Extreme: Extreme-Netlogin-VLAN-Name = purple
```

VSA 204: Extreme-Netlogin-URL

The Extreme-*NetLogin*-Url attribute specifies a web page URL that the RADIUS server sends to the switch after successful authentication. When the switch receives the attribute in response to a web-based network login, the switch redirects the web client to display the specified web page. If a login method other than web-based is used, the switch ignores this attribute.

The following describes the guidelines for VSA 204:

- To specify the URL to display after authentication, use an ASCII string.
- If you do not specify a URL, the network login infrastructure uses the default redirect page URL, , or the URL that you configured using the `configure netlogin redirect-page` command.
- VSA 204 applies only to the web-based authentication mode of Network Login.

The following example specifies the redirection URL to use after successful authentication.

To configure the redirect URL as `http://www.myhomepage.com`, add the following line:

```
Extreme: Netlogin-URL = http://www.myhomepage.com
```

VSA 205: Extreme-Netlogin-URL-Desc

The Extreme-NetLogin-Url-Desc attribute provides a redirection description that the RADIUS server sends to the switch after successful authentication. When the switch receives this attribute in response to a web-based network login, the switch temporarily displays the redirect message while the web client is redirected to the web page specified by attribute 204. If a login method other than web-based is used, the switch ignores this attribute.

The following describes the guidelines for VSA 205:

- To let the user know where they will be redirected to after authentication (specified by VSA 204), use an ASCII string to provide a brief description of the URL.
- VSA 205 applies only to the web-based authentication mode of Network Login.

The following example specifies a redirect description to send to the switch after successful authentication:

```
Extreme: Netlogin-URL-Desc = "Authentication successful. Stand by for the home page."
```

VSA 206: Extreme-Netlogin-Only

The Extreme-Netlogin-Only attribute can be used to allow normal authentication or restrict authentication to only the network login method.

When this attribute is assigned to a user and authentication is successful, the RADIUS server sends the configured value back to the switch. The configured value is either disabled or enabled.

The Extreme switch uses the value received from the RADIUS server to determine if the authentication is valid. If the configured value is disabled, all normal authentication processes are supported (Telnet and SSH, for example), so the switch accepts the authentication. If the configured value is enabled, the switch verifies whether network login was used for authentication. If network login was used for authentication, the switch accepts the authentication. If an authentication method other than network login was used, the switch rejects the authentication.

Add the following line to the RADIUS server users file for users who are not restricted to network login authentication:

```
Extreme:Extreme-Netlogin-Only = Disabled
```

Add the following line to the RADIUS server users file for users who are restricted to network login authentication:

```
Extreme:Extreme-Netlogin-Only = Enabled
```

To reduce the quantity of information sent to the switch, the RADIUS server sends either a 1 for the enabled configuration or a 0 for the disabled configuration.

These values must be configured in the RADIUS dictionary file as shown in Configuring the Dictionary File on page 1626.

VSA 209: Extreme-Netlogin-VLAN-ID

This attribute specifies a destination VLAN ID (or VLAN tag) that the RADIUS server sends to the switch after successful authentication.

The VLAN must already exist on the switch. When the switch receives the VSA, it adds the authenticated user to the VLAN.

The following describes the guidelines for VSA 209:

- For untagged VLAN movement with 802.1X netlogin, you can use all current Extreme Networks VLAN VSAs: VSA 203, VSA 209, and VSA 211.
- To specify the VLAN ID, use an ASCII string.
- When using this VSA, do not specify whether the VLAN is tagged or untagged.

Because the RADIUS server can identify a target VLAN with multiple attributes, the switch selects the appropriate VLAN or VLANs using the order:

- Extreme-Netlogin-Extended-VLAN (VSA 211)
- Extreme-Netlogin-VLAN-Name (VSA 203)
- Extreme-Netlogin-VLAN-ID (VSA 209)
- Tunnel-Private-Group-ID, but only if Tunnel-Type == VLAN(13) and Tunnel-Medium-Type == 802 (6) (see Standard RADIUS Attributes Used by Extreme Switches on page 1616)

If none of the previously described attributes are present ISP mode is assumed, and the client remains in the configured VLAN.

When added to the RADIUS users file, the following example specifies the destination VLAN ID, 234, for the associated user:

```
Extreme:Extreme-Netlogin-VLAN-ID = 234
```

VSA 211: Extreme-Netlogin-Extended-Vlan

This attribute specifies one or more destination VLANs that the RADIUS server sends to the switch after successful authentication.

You can specify VLANS by VLAN name or ID (tag). The VLANs may either already exist on the switch or, if you have enabled dynamic VLANs and a non-existent VLAN tag is given, the VLAN is created.

When the switch receives the VSA, it does the following:

- Unauthenticates the user on all VLANs where it is currently authenticated during reauthentication.
- Authenticates the user on all VLANs in the VSA, or none of them.

In cases where the client is already authenticated, if a single VLAN move fails from a list of VLANs in the VSA and the move-fail-action is authenticate, then it is left as-is. If the client is not already authenticated (first time authentication), then it is authenticated on learnedOnVlan if possible. If move-fail-action is deny then the client is unauthenticated from all the VLANs where it is currently authenticated. There is no partial success.

Note
If there is one or more invalid VLAN in the VSA, the supplicant is not authenticated on any one of them.

For example, if the VSA is Uvoice;Tdata and the VLAN data does not have a tag or the VLAN does not exist, then the port movement fails. Even if a single VLAN in the list is invalid the entire list is discarded and the action taken is based on move-fail-action configuration.

The following describes the guidelines for VSA 211:

- For tagged VLAN movement with 802.1X netlogin, you must use VSA 211.
- To specify the VLAN name or the VLAN ID, use an ASCII string; however, you cannot specify both the VLAN name and the VLAN ID at the same time. If the string only contains numbers, it is interpreted as the VLAN ID.
- A maximum of 10 VLANs are allowed per VSA.
- For tagged VLANs, specify **T** for tagged before the VLAN name or VLAN ID.
- For untagged VLANs, specify **U** for untagged before the VLAN name or VLAN ID.
- For movement based on the incoming port's traffic, specify the wildcard * before the VLAN name or VLAN ID. The behavior can be either tagged or untagged, based on the incoming port's traffic, and mimics the behavior of VSA 203 and VSA 209, respectively.
- Multiple VLAN names or VLAN IDs are separated by semicolons. When multiple vlans are defined in single VSA 211, the wildcard * is not allowed.
- There cannot be more than one untagged VLAN in a single VSA.
- The same VLAN cannot be both untagged and tagged in a single VSA.
- A client or supplicant can be authenticated in a only one untagged VLAN.
- The ports configured for an untagged VLAN different from the netlogin VLAN can never be added tagged to the same VLAN.
- A port can be in more than one untagged VLAN when MAC-based VLANs are enabled.

When added to the RADIUS users file, the following examples specify VLANs for the switch to assign after authentication:

```
Extreme-Netlogin-Extended-VLAN = Tvoice (Tagged VLAN named voice)
Extreme-Netlogin-Extended-VLAN = Udata (Untagged VLAN named data)
Extreme-Netlogin-Extended-VLAN = *orange (VLAN named orange, tagging dependent on
incoming traffic)
Extreme-Netlogin-Extended-VLAN = T229 (Tagged VLAN with ID 229)
Extreme-Netlogin-Extended-VLAN = U4091 (Untagged VLAN with ID 4091)
Extreme-Netlogin-Extended-VLAN = *145 (VLAN with ID 145, tagging dependent on incoming
traffic)
in FreeRADIUS, a tagged VLAN voice and a tagged VLAN mktg would be configured as the
following:
Extreme-Netlogin-Extended-VLAN = "Tvoice;Tmktg;"
```

An untagged VLAN data and a tagged VLAN mktg is configured as the following:

```
Extreme-Netlogin-Extended-VLAN = "Udata;Tmktg;"
```

A tagged VLAN with VLAN ID 229 and a tagged VLAN with VLAN ID 227 is configured in FreeRADIUS as the following:

```
Extreme-Netlogin-Extended-VLAN = "T229;T227;"
```

An untagged VLAN with VLAN ID 4091 and a tagged VLAN with VLAN ID 2001 is configured as the following:

```
Extreme-Netlogin-Extended-VLAN = "U4091;T2001;"
```

VSA 212: Extreme-Security-Profile

This attribute specifies a profile name that the RADIUS server sends to the switch after successful authentication. The switch uses this profile name to run a special type of script called a *profile*. The profile is stored on the switch and can be used to modify the switch configuration in response to authentication. Profiles are created using the Universal Port feature, which is described in Universal Port on page 2135.

The following describes the guidelines for VSA 212:

- This VSA must contain a profile name.
- This VSA can include optional variables for use in profile execution.
- The variable entry format is: `<var1>=<value1>;<var2>=<value2>;...`
- Each profile variable must be separated from the others by a semicolon.

When added to the RADIUS users file, the following example provides to the switch the profile name p1, variable QOS=QP8, and variable LOGOFF-PROFILE=P2:

```
EXTREME-SECURITY-PROFILE= "p1 QOS=\"QP8\";LOGOFF-PROFILE=P2;"
```

VSA 213: EXTREME_VM_NAME

This VSA is used in context with the *Extreme Network Virtualization (XNV)* feature, especially with the NMS authentication of VMs. Use this VSA to specify the name of the VM that is being authenticated. An example would be: MyVM1

VSA 214: EXTREME_VM_VPP_NAME

This VSA is used in context with the XNV feature, especially with the NMS authentication of VMs. Use this VSA to specify the VPP to which the VM is to be mapped. An example would be: nvpp1

VSA 215: EXTREME_VM_IP_ADDR

This VSA is used in context with the XNV feature, especially with the NMS authentication of VMs. Use this VSA to specify the IP address of the VM. An example would be: 11.1.1.254

VSA 216: EXTREME_VM_VLAN_ID

This VSA corresponds to XNV with Dynamic VLANs. Use this VSA to specify the ID or tag of the destination VLAN for the VM. An example would be: 101

VSA 217: EXTREME_VM_VR_NAME

This VSA corresponds to XNV with Dynamic VLANs. Use this VSA to specify the VR in which the destination VLAN is to be placed. An example would be: UserVR1

Configuring the Dictionary File

Before you can use Extreme Networks VSAs on a *RADIUS* server, you must define the VSAs.

On the FreeRADIUS server, you define the VSAs in the dictionary file in the /etc/raddb directory. You must define the vendor ID for Extreme Networks, each of the VSAs you plan to use, and the values to send for the VSAs. The following example shows the entries to add to a FreeRADIUS server dictionary file for Extreme Networks VSAs:

```
VENDOR          Extreme         1916
ATTRIBUTE    Extreme-CLI-Authorization      201  integer     Extreme
ATTRIBUTE    Extreme-Shell-Command          202   string     Extreme
ATTRIBUTE    Extreme-Netlogin-Vlan          203   string     Extreme
ATTRIBUTE    Extreme-Netlogin-Url           204   string     Extreme
ATTRIBUTE    Extreme-Netlogin-Url-Desc      205   string     Extreme
ATTRIBUTE    Extreme-Netlogin-Only          206  integer     Extreme
ATTRIBUTE    Extreme-User-Location          208   string     Extreme
ATTRIBUTE    Extreme-Netlogin-Vlan-Tag      209  integer     Extreme
ATTRIBUTE    Extreme-Netlogin-Extended-Vlan 211   string     Extreme
ATTRIBUTE    Extreme-Security-Profile       212   string     Extreme
ATTRIBUTE    Extreme-Policy-ACL             232   string     Extreme
VALUE        Extreme-CLI-Authorization   Disabled       0
VALUE        Extreme-CLI-Authorization   Enabled        1
VALUE        Extreme-Netlogin-Only       Disabled       0
VALUE        Extreme-Netlogin-Only       Enabled        1
# End of Dictionary
```

The lines that begin with VALUE provide the integers that the RADIUS server sends to the switch when the corresponding text is configured in the RADIUS users file. For example, if the Extreme-CLI-Authorization attribute is set to Enabled for a particular user, the RADIUS server sends the value 1 to the switch (which reduces total bytes transferred). The ExtremeXOS software is designed to interpret the integer values as shown above, so be sure to use these values.

Change-of-Authorization (Dynamic Authorization) Overview

The *RADIUS* protocol, defined in (RFC2865), does not support unsolicited messages sent from the RADIUS server to the Network Access Server (NAS). However, it may be desirable for changes to be made to session characteristics, without requiring the NAS to initiate the exchange. For example, it may be desirable for administrators to be able to terminate user session(s) in progress. Alternatively, if the user changes authorization level, this may require that authorization attributes be added/deleted from user session(s). To overcome these limitations, several vendors have implemented additional RADIUS commands to enable unsolicited messages to be sent to the NAS. These extended commands provide support for Disconnect and Change-of-Authorization (CoA) packets.

Disconnect Request

Disconnect Requests are sent from the DA initiator to the DA controller when it is determined that a session will be terminated. The request is sent to UDP port 3799.

When a Disconnect Request packet is received, it is determined whether it is an authorized request from a configured server. The source IP address of the packet is used to check for a match of the configured Dynamic Authorization Servers. If the source IP address has not been configured, then the request is immediately dropped with no further validation. If the IP address is present in the configuration, a validation of the Message-Authenticator attribute occurs as indicated in RFC2869. If validation is not successful, then the packet is dropped with no further processing. If it is determined that this is a retry, the packet is also dropped with no further processing.

The authorized request indicates which DA controller the message is for, as well as which session should be terminated. If the DA controller indicated by the included attributes in the packet does not match the receiver, then the request is responded to with a Disconnect-NAK. The appropriate DA controller receiving this *RADIUS* extension packet identifies the session using the attributes provided and immediately terminates the session.

Note
In order to use CoA/Disconnect, ONEPolicy must be enabled with NetLogin.

NAS Identification

These attributes provided by the *RADIUS* extension packets are used to determine the DA Controller that is to disconnect the session:

- NAS-IP-Address—This IPV4 address must match the primary IP address of the default interface for a match to occur.

If all of these attributes do not match, the request is responded to with a Disconnect-NAK response.

Starting with ExtremeXOS 31.3, the nas-ip option can ignore this requirement using the `ignore` variable.

User Session Identification

The following attributes are used to determine the Layer 2 sessions that are to be terminated:

- NAS-Port—the interface index of the port that the session is connected to.
- Calling-Station-Id—The ASCII representation of the MAC address of the session to be terminated. If no NAS-Port attribute is present, then all sessions with the specified MAC address are terminated regardless of what port(s) the sessions are connected to.
- Enterasys Auth-Client-Type—One or more of these vendor-specific attributes (VSA) can be specified to indicate which authentication clients are to be affected. Valid values are: 1-dot1x, 2-pwa, 3-macauth. If this attribute is not present, then all sessions with the specified MAC address and/or port are affected.

All included attributes must match for the request to be processed. Additional attributes that are not understood by the DA controller in this context cause a failure of this disconnect request.

Disconnect Responses

If the Disconnect Request is determined to be of appropriate format and matching session(s) are present and terminated appropriately, then a Disconnect-Ack is sent in response. If all the attributes in the packet are not understood within this context or do not result in a matching DA controller and session, then the DA controller sends a Disconnect-NAK in response. If the DA controller is unable to appropriately terminate the matching session(s) for some reason, a Disconnect-NAK is sent in response.

Security

Using the *RADIUS* protocol to push both changes and terminations of existing sessions does have some security issues.

Caution
In this implementation in ExtremeXOS of RFC5176, some level of trust is required on the upstream portion of the local network. Network administrators should not enable this functionality if they are not comfortable with this level of security. See Limitations on page 527.

Limitations

The following features of Change-of-Authorization (RFC5176) are not implemented in ExtremeXOS:

- Reverse Path Forwarding Check—Typically this is used in a proxy scenario. This check is used to determine if the IP address indicated by the *RADIUS* attributes is a routable destination address for a request sent by the switch software.
- IPSEC encryption—End-to-end encryption of both the RADIUS requests and responses.
- Disconnect-Request and Change-of-Authorization packets identifying sessions with anything other than the Calling-Station-Id attribute containing a properly formatted MAC address. In addition to the Calling-Station-ID attribute, you can also use a NAS-

Port attribute, which indicates the index of the specific port the session is connected to.

- Acct-Session-Id attribute—This is an alternate means of session identification. Sessions are currently uniquely identified by port and MAC address pair.
- Retransmissions of Disconnect-Request or Change-of-Authorization ACK and NAK packets—Retransmissions of packets is the responsibility of the device initiating the dynamic authorization transactions.

Change of Authorization

Change of Authorization requests are sent from the DA initiator to the DA controller when it is determined that a session's authorization parameters will be changed. The request is sent to UDP port 3799.

When a Change-of-Authorization (CoA) packet is received, it is determined whether it is an authorized request from a configured server. The source IP address of the packet is used to check for a match of the configured Dynamic Authorization Servers. If the source IP address has not been configured, then the request is immediately dropped with no further validation. If the IP address is present in the configuration, a validation of the Message-Authenticator attribute occurs as indicated in RFC2869. If validation is not successful, then the packet is dropped with no further processing. If it is determined that this is a retry, the packet is also dropped with no further processing.

The message indicates which DA controller the message is for, as well as, which session should be terminated. If the DA controller indicated by the included attributes in the packet does not match the receiver, then the request is responded to with a Change-of-Authorization-NAK. The appropriate DA controller receiving this *RADIUS* extension packet identifies the session(s) using the attributes provided and attempts to change the authorization level of the user as newly defined in the request packet.

DA Controller Identification

The DA controller is identified for CoA Requests in the same way that Disconnect Request identification occurs.

User Session Identification

Session identification for CoA occur in the same way that Disconnect Request identification does.

Supported Authorization Attributes

Policy authorization levels can be changed using this functionality. These attributes are the same ones that are used to define the initial authorization level of a session when authenticated by the DA controller. The processing of these attributes will result in the same authorization conditions that would have occurred if they were to have been received as the initial authorization level.

- Filter-Id —The filter ID identifies the name of the policy profile that is used for the session.
- RFC3580 Attributes—Three attributes are used to determine the RFC3580 authorization levels: Tunnel-Type, Tunnel-Medium-Type, and Tunnel-Private-Group-Id.

These values are used in concert with local policy settings to determine both the correct policy profile and *VLAN* settings for the session.

Change-of-Authorization Responses

If the CoA request is determined to be of appropriate format, matching session(s) are found, and their authorization levels are changed appropriately, then a Change-Of-Authorization-Ack is sent in response. If all the attributes in the packet are not understood within this context or do not result in a matching DA controller and session, then the DA controller sends a Change-Of-Authorization-NAK in response. If the DA controller is unable to appropriately change the matching session(s) authorization level for some reason, a Change-Of-Authorization-NAK is sent in response.

Retry Detection

To minimize the potential for replay attacks, the DA controller needs to support retry detection. This involves remembering source UDP port and *RADIUS* packet identifier pairs that have been received on a per DA Initiator basis. This information needs to be remembered for a reasonable period of time so that an attacker resending an identical frame to the DA controller does not cause undue harm to the switch. This functionality used in conjunction with the Event-Timestamp attribute limits the validity of replay attacks.

Change-of-Authorization Mandatory Attributes

The following is a list of Change-of-Authorization CoA mandatory attributes.

- Calling-Station-ID = "session-id Hyponated"
- NAS-IP-Address = <SWITCH MGMT IP>
- Filter-ID = <Policy Profile Name>
- Tunnel-Medium-Type = IEEE-802
- Tunnel-Type = VLAN
- Tunnel-Private-Group-Id = <Tunnel-ID>

Note
Filter-ID and Tunnel attributes are mandatory based on the configuration used in the switch. Filter-ID is mandatory if the map table response is policy. Tunnel is mandatory if map table response is tunnel. Attributes shown in Table 147 on page 1631 are optional attributes.

Limitations

Optional attributes may or may not be present in the CoA/disconnect. If all the received values are correct, then further processing occurs; otherwise, you may get an ACK/NAK, but the packet is not processed further.

New Attribute Support

Table 147: New CoA Attribute Support

Name	Description	Defining RFC
Event-Timestamp	The Event-Timestamp attribute is used to minimize the effect of network replay attacks. RFC5176 recommends incorporating this attribute when not using more complex security measures to encrypt the *RADIUS* packet data. The DA controller does not process Disconnect Request or CoA requests that do not include this attribute. The timestamp sent in this attribute must be within 300 seconds of the current time for the request to be processed. Response frames to either Disconnect request or CoA requests contain this attribute.	RFC2869
Proxy-State	When one or more of these attributes are included in either Disconnect request or request frames they must be included unedited in the responses to those packets.	RFC2865
Message-Authenticator	The Message-Authenticator attribute is used to both authenticate and integrity check RADIUS packets. It is used in lieu of more complex security measures to authorize and/or encrypt the RADIUS control packets. The DA controller does not process packets with invalid Message-Authenticator attribute values.	RFC2869
Error-Cause	The Error-Cause attribute is used to give the DA Initiator more information regarding the cause of the failure to process either a Disconnect request or a CoA request. The DA controller uses this attribute when it responds with the Disconnect-Request-NAK or the Change-Of-Authorization-NAK messages.	RFC5176

Table 147: New CoA Attribute Support (continued)

Name	Description	Defining RFC
Enterasys Auth-Client-Type	The Enterasys Auth-Client-Type vendor-specific attribute (VSA) is used to indicate which authentication client sessions are to be affected by either the CoA or Disconnect Requests. The vendor ID used for this VSA is the IANA assigned private enterprise number for Enterasys—5624. The Enterasys attribute type number for this attribute is 1. Valid values are: 1- dot1x, 2-pwa, 3-macauth, 4-cep, 5-radsnoop, 6-auto-tracking and 7-quarantine-agent.	N/A
NAS-Port	The interface index of the port that a session is connected to. Representation of port 5 in standalone switches will be like this: 1005, stack port 3:5 will be 3005, and vpex port 100:5 will be like this NAS-Port = 101005.	RFC2865
Extreme-Policy-ACL	See Dynamic ACLs on page 370.	EXTREME-VSA 232 String
Extreme-Port-Bounce	Specifies whether to bounce (disable/enable) the port during authenticationValues: • 0 = No (do not bounce port) • 1 = Yes (bounce port) Usage: Sent by RADIUS server in Access-Accept messages to trigger port bounce operation	EXTREME-VSA 233 String

RADIUS Port Bounce

Port bouncing is the process of temporarily disabling and re-enabling a network port, which enables the clients connected to an authenticator to reinitiate a DHCP request. RADIUS port bounce occurs only when there is a change in the authenticator VLAN and when a CoA request is received from a RADIUS server with VSA port bounce. You can enable or disable port bounce using the command line interface, but first you must do the following:

- Start an active Netlogin session on an authenticator port.
- Create Policy.
- Enable Dynamic Authentication.

The following command configures the port bounce feature:

```
configure radius port-bounce [ on | off]
```

Additional RADIUS Configuration Examples

Installing and Testing the FreeRADIUS Server

RADIUS is a client/server protocol based on UDP.

The example presented in this section describes a RADIUS server that is a daemon process running on a Linux server.

The following example shows how to install and test a FreeRADIUS server:

```
tar -zxvf freeradius-1.0.2.tar.gz        (extract with gunzip and tar)
./configure
make
make install                             (run this command as root)
radiusd                                  (start RADIUS server, or...)
radiusd -X                               (start RADIUS server in debug mode)
radtest test test localhost 0 testing123 (test RADIUS server)
```

If radtest receives a response, the FreeRADIUS server is up and running.

Note
RADIUS server software can be obtained from several sources. This solution uses the FreeRADIUS software available on the following URLs: www.freeradius.org and www.redhat.com. Another free tool, NTRadPing, can be used to test authentication and authorization requests from Windows clients. NTRadPing displays detailed responses such as attribute values sent back from the RADIUS server.

Configuring the FreeRADIUS Server

Configuring the *RADIUS* server involves configuring the RADIUS server and the RADIUS client (for authentication and authorization). FreeRADIUS configuration files are usually stored in the /etc/raddb folder. The following example demonstrates how to configure the FreeRADIUS server for authentication and LDAP support:

1. Modify the radiusd.conf file global settings:

```
log_auth = yes              (log authentication requests to the log file)
log_auth_badpass = no       (don't log passwords if request rejected)
log_auth_goodpass = no      (don't log passwords if request accepted)
```

2. Modify LDAP Settings:

```
modules {
            ldap {
                server = "ldaptest.extremenetworks.com"
                basedn = "o=ldaptestdemo,dc=extremenetworks,dc=com"
                filter = "(cn=%{Stripped-User-Name:-%{User-Name}})"
                base_filter = "(objectclass=radiusprofile)"
                start_tls = no
                dictionary_mapping = ${raddbdir}/ldap.attrmap
                authtype = ldap
                ldap_connections_number = 5
                timeout = 4
        timelimit = 3
        net_timeout = 1
                }
}
```

3. Uncomment LDAP from the authorize section:

```
               authorize {
            preprocess
            chap
            mschap
            suffix
            ldap
            eap
            files
                }
```

4. Uncomment LDAP from the authenticate section:

```
        authenticate {
        Auth-Type PAP {
        pap
            }
        Auth-Type CHAP {
        chap
            }
        Auth-Type MS-CHAP {
        mschap
            }
            unix
        ldap
            eap
```

   An Extreme Networks edge switch serves as a network access server (NAS) for workstations and as a RADIUS client for the RADIUS server.

   RADIUS clients are configured in /etc/raddb/clients.conf. There are two ways to configure RADIUS clients. Either group the NAS by IP subnet or list the NAS by host name or IP address.

5. Configure the RADIUS client using the second method.

```
client 192.168.1.1 {
          secret = extreme1
          shortname = ldap-demo
}
```

Configuring the RADIUS-to-LDAP Attribute Mappings

Attributes are configured in /etc/freeradius/ladp.attrmap. This file maps *RADIUS* attributes to LDAP attributes. Samba has NT/LM password hashes. Hence, the default mapping for LM-Password and NT-Password must be changed.

- Configure attribute mappings.

```
checkItem User-Password userPassword.
checkItem LMPassword sambaLMPassword
checkItem NTPassword sambaNTPassword
replyItem Tunnel-Type radiusTunnelType
replyItem Tunnel-Medium-Type radiusTunnelMediumType
replyItem Tunnel-Private-Group-Id radiusTunnelPrivateGroupId
```

Configuring Additional Attributes Mappings

Attributes are configured in /etc/freeradius/ladp.attrmap:

```
## Attributes for Extreme Networks Vendor-Specific RADIUS
replyItem   Extreme-Security-Profile radiusExtremeSecurityProfile
```

```
replyItem   Extreme-Netlogin-Vlan-Tag radiusExtremeNetloginVlanTag
replyItem   Extreme-Netlogin-Extended-Vlan radiusExtremeNetloginExtendedVlan
```

Modifying the RADIUS Schema

Additional attributes for *RADIUS* must be configured to extend the RADIUS-LDAP-V3.schema under the /etc/openldap directory.

- Use the following commands to modify the RADIUS schema:

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.61
NAME 'radiusExtremeSecurityProfile'
DESC ''
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)
attributetype
( 1.3.6.1.4.1.3317.4.3.1.62
NAME 'radiusExtremeNetloginVlanTag'
DESC ''
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)
attributetype
( 1.3.6.1.4.1.3317.4.3.1.63
NAME 'radiusExtremeNetloginExtendedVlan'
DESC ''
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)
```

Configuring the Authentication Method for Supplicants

The authentication method is configured in /etc/raddb/eap.conf. The authentication method used by FreeRADIUS is the PEAP (Protected EAP) method. To activate PEAP, a TLS tunnel is required to encrypt communication between supplicant and *RADIUS* server. This means that server certificates are required.

- Configure the authentication method.

```
peap {
default_eap_type = mschapv2
}
tls {
private_key_password = whatever
private_key_file = ${raddbdir}/certs/cert-srv.pem
certificate_file = ${raddbdir}/certs/cert-srv.pem
CA_file = ${raddbdir}/certs/demoCA/cacert.pem
dh_file = ${raddbdir}/certs/dh
random_file = ${raddbdir}/certs/random
fragment_size = 1024
include_length = yes
}
```

Starting the FreeRADIUS Server

- Start *RADIUS* in the foreground with debugging enabled.

  ```
  radiusd -X -f
  ```

Refresh RADIUS TLS After Certificate/Key Update

Version 33.4.1 adds support for dynamically updating the TLS certificate used by RadSec without requiring a full unconfiguration and reconfiguration of the RADIUS service.

To apply a newly imported certificate/key pair for RADIUS TLS without reconfiguring all RADIUS servers, use the following command:

```
refresh radius tls
```

Implementation Notes for Specific RADIUS Servers

Cistron RADIUS

Cistron Radius is a popular server, distributed under GPL. Cistron Radius can be found at:

When you configure the Cistron server for use with Extreme switches, you must pay close attention to the users file setup. The Cistron Radius dictionary associates the word Administrative-User with Service-Type value 6, and expects the Service-Type entry to appear alone on one line with a leading tab character.

The following is a user file example for read-write access:

```
adminuser   Auth-Type = System
Service-Type = Administrative-User,
Filter-Id = “unlim”
```

RSA Ace

For users of Cistron's RSA SecureID® product, RSA offers *RADIUS* capability as part of their RSA/Ace Server® server software. It is mandatory to configure a matching shared-secret key on the switch and RSA Ace server for successful authentication.

Steel-Belted Radius

For users who have the Steel-Belted Radius (SBR) server from Juniper Networks, it is possible to limit the number of concurrent login sessions using the same user account. This feature allows the use of shared user accounts, but limits the number of simultaneous logins to a defined value. Using this feature requires Steel-Belted Radius for *RADIUS* authentication and accounting. To limit the maximum concurrent login sessions under the same user account:

1. Configure RADIUS and RADIUS-Accounting on the switch.

   The RADIUS and RADIUS-Accounting servers used for this feature must reside on the same physical RADIUS server. Standard RADIUS and RADIUS-Accounting configuration is required as described earlier in this chapter.
2. Modify the SBR vendor.ini file and user accounts.
   a. To configure the SBR server, the file vendor.ini must be modified to change the Extreme Networks configuration value of ignore-ports to yes as shown in the example below:

```
vendor-product       = Extreme Networks
dictionary           = Extreme
ignore-ports         = yes
```

```
port-number-usage     = per-port-type
help-id               = 2000
```

b. After modifying the vendor.ini file, the desired user accounts must be configured for the Max-Concurrent connections. Using the SBR Administrator application, enable the check box for Max-Concurrent connections and fill in the desired number of maximum sessions.

Microsoft IAS

To use Extreme Networks VSAs with the Internet Authentication Service (IAS) in Microsoft® Windows Server™ 2003, you must first create a Remote Access Policy and apply it so that user authentication occurs using a specific authentication type such as EAP-TLS, PEAP, or PAP. The following procedure assumes that the Remote Access Policy has already been created and configured and describes how to define Extreme Networks VSAs in Microsoft IAS:

1. Open the IAS administration GUI application.
2. In the left window pane, select the **Remote Access Policies** section of the tree.
3. In the right window pane, double-click the desired Remote-Access policy name so you can edit it.
4. Click the **Edit-Profile** button in the lower-left corner, and then select the **Advanced** tab.
5. If any attributes already appear in the **Parameters** window, remove them by selecting the attribute and clicking the **Remove** button.
6. When the **Parameters** window is empty, proceed to the next step.
7. Click the **Add** button, which brings up the **Add Attributes** dialog window.
8. Scroll down the displayed list of *RADIUS* attributes and select the attribute named **Vendor-Specific**.
9. Double-click the **Vendor-Specific** attribute or click the **Add** button.

   The **Multivalued Attribute Information** dialog box should appear.
10. Click the **Add** button, which brings up the **Vendor-Specific Attribute Information** dialog window.
    a. Select the first radio button for **Enter Vendor Code** and enter the Extreme Networks vendor code value of `1916` in the text-box.
    b. Select the second radio button for **Yes, It conforms**.
    c. Verify both settings, and click the **Configure Attribute** button to proceed.

       The **Configure VSA (RFC compliant)** dialog window should now appear.

       The settings for this dialog window varies, depending on which product and attribute you wish to use in your network.
    d. In the first text-box enter the Extreme Networks VSA number for the attribute you want to configure (see Extreme Networks VSAs on page 1619).
    e. Use the pull-down menu to select the **Attribute** format, which is the same as the attribute Type listed in Extreme Networks VSAs on page 1619.

Note
For values of format integer you will have to select the type **Decimal** from the pull-down menu.

f. Configure the desired value for the attribute.
g. Once the desired values have been entered, click **OK**.

11. Click **OK** two more times to return to the **Add Attributes** dialog window.
12. Select **Close**, and then click **OK** twice to complete the editing of the Remote Access Policy profile.
13. To apply the configuration changes, stop and restart the Microsoft IAS service.

    After restarting the IAS service, new authentications should correctly return the Extreme Networks VSA after successful authentication. Users who were previously authenticated have to re-authenticate to before the new VSAs apply to them.
14. If you experience problems with the newly configured VSAs, use the following troubleshooting guidelines:
    a. If you have multiple IAS Remote Access Policies, verify that the user is being authenticated with the correct policy.
    b. Check the IAS System Log events within Microsoft Event Viewer to verify the user is authenticated through the policy where VSA settings are configured.
    c. Check whether the VSA configuration performed above is correct.

       A mismatch in any of the VSA settings could cause authentication or VSA failure.
    d. Verify that attributes such as "*VLAN* tag" or "VLAN name" correctly match the configuration of your ExtremeXOS switch and overall network topology.

       Invalid, or incorrect values returned in the VSA could prevent authenticated users from accessing network resources.

Setting Up Open LDAP

To integrate an ExtremeXOS switch in an LDAP environment, a *RADIUS* server must be configured to communicate with the LDAP database. The following components are required to install the access control solution:

- Linux server with Linux Red Hat 4.0
- FreeRADIUS 1.1.x
- OpenLDAP 2.3.x
- Extreme Networks switches
- Windows 7/Windows 8 clients

To configure Universal Port for use in an LDAP environment, use the following procedure:

1. Install and configure a RADIUS server on an existing Linux server as described in Installing and Testing the FreeRADIUS Server on page 1633.
2. Install and configure OpenLDAP as described later in this section.
3. Add vendor specific attributes to the RADIUS and LDAP servers as described in Installing and Testing the FreeRADIUS Server on page 1633 and later in this section.
4. Configure the edge switches as described in this guide.
5. Configure each supplicant as described in Configuring a Windows 7/Windows 8 Supplicant for 802.1X Authentication on page 1643.

For complete instructions on setting up an LDAP server, see the product documentation for the LDAP server.

Installing OpenLDAP

OpenLDAP software is an open source implementation of Lightweight Directory Access Protocol. This can be obtained from the site: www.openldap.org. To install OpenLDAP packages:

1. Verify the Red Hat Linux installed releases.

   The release number is stored in the /etc/redhat-release file.
2. Verify the version of OpenLDAP currently installed by entering the command `rpm -qa | grep openldap` at the Linux prompt.

   ```
   # rpm -qa |grep openldap
   openldap-2.3.xx-x
   openldap-clients-2.3.xx-x
   openldap-servers-2.3.xx-x
   ```
3. If you have a default Red Hat Linux installation, there is at least one OpenLDAP Red Hat Package Manager (RPM) installed.

   The LDAP RPMs can be found on the Red Hat CD or downloaded from one of the following RPM download sources:

   www.rpmfind.net and search for **openldap** and select the RPM based on the distribution

   www.redhat.com and select **Download**, and then search for **openldap**.
4. After downloading the RPMs to the Linux server, change to the download directory and start the installation using the rpm command:

   `# rpm -ivh openldap*`
5. Verify that the OpenLDAP RPMs have been installed with the `rpm -qa | grep openldap` command at the Linux prompt.

   ```
   # rpm -qa | grep openldap
   openldap-2.3.xx-x
   openldap-clients-2.3.xx-x
   openldap-servers-2.3.xx-x
   ```

Configuring OpenLDAP

Once the build is complete, the slapd and slurpd daemons are located in /usr/local/libexec.

The config files are in /etc/openldap and ready to start the main server daemon, slapd.

Configuring slapd for Startup

Before you start slapd, edit /etc/openldap/slapd.conf to include the location to store the data and details on who is allowed to access the data.

The following configuration changes need to be made:

- Change the suffix.
- Change the rootDN.
- Use slappasswd to generate rootpw.
- Add rootpw entry.

Use the following commands to configure slapd for startup:

```
database (use default)
suffix "dc=xxxxxx,dc=org"
rootdn "cn=xxxx,dc=xxxxxx,dc=org"
rootpw {SSHA}c5PemO1KWqz0254r4rnFVmxKA/evs4Hu
directory /var/lib/ldap
allow bind_v2
pidfile /var/run/slapd.pid
```

Adding New Schemas

The *RADIUS* schema and Samba schema for PEAP authentication must be included into the slapd.conf file. After modifying the file, the LDAP server must be restarted to load the new schemas.

- Use the following commands to add new schemas:

```
cp /usr/share/doc/freeradius-1.0.1/RADIUS-LDAPv3.schema /etc/openldap/schema/
```

```
cp /usr/share/doc/samba-3.0.10/LDAP/samba.schema /etc/openldap/schema
```

- Use the following commands to modify slapd.conf:

```
include/etc/openldap/schema/RADIUS-LDAPv3.schema
include/etc/openldap/schema/samba.schema
```

Populating the LDAP Database with Organization and User Entries

- Use the following commands to make the user entry in the LDAP directory (slapd.conf):

```
dn: uid=newperson3,o=ldaptestdemo,dc=extremenetworks,dc=com
objectClass: top
objectClass: person
objectClass: radiusprofile << Defined in the RADIUS-LDAPv3 schema
objectClass: sambaSamAccount
sn: ldaptestdemo
uid: newperson3 <<< This username given in the Odyssey client
cn: newperson3
radiusTunnelMediumType: IEEE-802
radiusTunnelType: VLAN
radiusTunnelPrivateGroupId: 2 <<< Value of the VLAN tag
sambaNTPassword: A3A685F89364D4A5182B028FBE79AC38
sambaLMPassword: C23413A8A1E7665FC2265B23734E0DAC
userPassword:: e1NIQX00MXZzNXNYbTRPaHNwUjBFUU9raWdxbldySW89
sambaSID: S-1-0-0-28976
```

The Samba-related attributes can be populated in the LDAP server already if there is an LDAP-enabled Samba infrastructure in place.

Note
If the Samba related entries are not present, then the values for sambaNTPassword and sambaNMPPassword can be created by running the `mkntpwd` command.

```
cd /usr/share/doc/samba-3.0.10/LDAP/smbldap-tools/mkntpwd
make
./mkntpwd -L <password>   (provides value for sambaLMPassword attribute)
./mkntpwd -N <password>   (provides value for sambaNTPassword attribute)
```

Restarting the LDAP Server

- Use the following syntax to stop and start LDAP services:

  ```
  service ldap restart
  ```

  For phone authentication (which uses EAP based md5 authentication), the password is stored in clear text in the UserPassword field for the phone entries in LDAP.

LDAP Configuration Example

This configuration example is for ExtremeSwitching switches

Use the following commands to activate the switch for 802.1X port-based authentication:

```
create vlan voice
create vlan data
create vlan ldap
configure voice tag 10
configure data tag 20
configure ldap ipaddress 192.168.1.1/24
cnable ipforwarding
create vlan nvlan
en netlogin dot1x
en netlogin port 13-24 dot1x
configure radius netlogin primary server 192.168.1.2 1812 client-ip 192.168.1.1 vr VR-
Default
configure radius netlogin primary shared-secret extreme1
enable radius netlogin
enable netlogin dot1x
```

Configure the ports to run a script when a user is authenticated through *RADIUS* and LDAP:

```
configure upm event user-authenticate profile a-avaya ports 1-23
LDAP UID entries:
```

In the LDAP phone UID entry in the users file, use the following attribute to specify a profile to run on the switch:

```
Extreme-Security-Profile
```

To add the port as tagged in the voice *VLAN*, use the following attribute in the users file:

```
Extreme-Netlogin-Extended-Vlan = TVoice (use UData for a PC)
```

Note
It depends on the end-station to determine the fields required for authentication; XP uses EAP-PEAP and must have encrypted fields for the UID password. Avaya phones authenticate with MD-5 and must have an unencrypted field in LDAP.

Scripts

The following a-avaya script tells the phone to configure itself in the voice VLAN, and to send tagged frames.

The script also informs the phone of the file server and call server:

```
create upm profile a-avaya
create log message Starting_UPM_Script_AUTH-AVAYA
set var callServer 10.147.12.12
set var fileServer 10.147.10.3
set var voiceVlan voice
set var CleanupProfile CleanPort
set var sendTraps false
#
create log message Starting_UPM_AUTH-AVAYA_Port_$EVENT.USER_PORT
#*********************************************************
# adds the detected port to the device "unauthenticated" profile port list
#*********************************************************
create log message Updating_Unauthenticated_Port_List_Port_$EVENT.USER_PORT
#configure upm event user-unauthenticated profile CleanupProfile ports $EVENT.USER_PORT
#*********************************************************
# Configure the LLDP options that the phone needs
#*********************************************************
configure lldp port $EVENT.USER_PORT advertise vendor-specific dot1 vlan-name vlan
$voiceVlan
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme call-server
$callServer
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme file-server
$fileServer
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme dot1q-
framing tagged
configure lldp port $EVENT.USER_PORT advertise vendor-specific med capabilities
#configure lldp port $EVENT.USER_PORT advertise vendor-specific med policy application
voice vlan $voiceVlan dscp 46
# If port is PoE capable, uncomment the following lines
#*************************************************************
# Configure the POE limits for the port based on the phone requirement
#*************************************************************
configure lldp port $EVENT.USER_PORT advertise vendor-specific med power-via-mdi
#configure inline-power operator-limit $EVENT.DEVICE_POWER ports $EVENT.USER_PORT
create log message UPM_Script_A-AVAYA_Finished_Port_$EVENT.USER_PORT
```

Note

Parts of the scripts make use of the QP8 profile. This is NOT recommended because the QP8 profile is used by *EAPS*. For voice, use QP7 for QOS.

This script uses tagging for the phone and the ports for the voice VLAN. This is NOT necessary; use multiple supplicant and untagged for the phones.

FreeRADIUS Sample Dictionary Values

```
Sample dictionary values for freeradius.
#
#       These attributes are examples
#

VENDOR          Enterasys               5624
BEGIN-VENDOR Enterasys
ATTRIBUTE       Enterasys-Auth-Client-Type      1       integer
VALUE   Enterasys-Auth-Client-Type      802.1X          1
VALUE   Enterasys-Auth-Client-Type      PWA             2
VALUE   Enterasys-Auth-Client-Type      MAC             3
END-VENDOR Enterasys

VENDOR          Extreme 1916
BEGIN-VENDOR Extreme
```

```
ATTRIBUTE       Extreme-Policy-ACL              232     string
END-VENDOR Extreme
```

Configuring a Windows 7/Windows 8 Supplicant for 802.1X Authentication

This section provides an overview procedure for configuring a Windows 7/Windows 8 supplicant. For complete instructions on setting up a Windows 7/Windows 8 supplicant, see the product documentation for Microsoft Windows XP.

Note
For enhanced security, install the FreeRADIUS server CA certificate (the CA that signed the certificate installed in eap.conf).

1. Open the network configuration panel, select the network card, enter the properties.
2. Click the **Authentication** tab.

   The Authentication dialog appears.
3. Enable 802.1X and disable authenticate as computer.
4. Choose EAP type of Protected EAP, then click **Properties**.
5. Deselect the Validate server certificate and select eap-mschapv2 as the authentication method.
6. Click **Configure**.
7. Select or clear the check box depending on whether you want to use the logon name and password, then click **OK**.

Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is a set of rules for transferring and exchanging information (data, voice, images, and so on) on the World Wide Web. HTTP is based on a request-response model. An HTTP client initiates requests by establishing a TCP connection to a port on a remote host (port 80 by default). An HTTP server listening on that port waits for and then responds to the request; in many instances, the client is requesting a specific URL or IP address. Upon receiving a request, the destination server sends back the associated file or files and then closes the connection.

The web server in ExtremeXOS allows HTTP clients to access the switch on port 80 (by default) as well as the network login page without additional encryption or security measures. For information about secure HTTP transmission, including Secure Socket Layer (SSL), see Secure Socket Layer on page 1676.

By default, HTTP is enabled on the switch.

- If you disabled HTTP access, you can re-enable HTTP access on the default port (80) using the following command:

  ```
  enable web http
  ```
- To disable HTTP, use the following command:

  ```
  disable web http
  ```

Starting in Switch Engine 32.7.1, HTTPS is also enabled by default under certain conditions so that both HTTP and HTTPS are active and available for use. HTTPS is enabled in::

- New switches with Switch Engine 32.7.1 installed.
- Old switches that are not configured and are booting with Switch Engine 32.7.1 (the switch does not have a configuration file).
- Old switches that are upgrading to Switch Engine 32.7.1 that have a configuration file where HTTPS is disabled and there is no switch key or certificate.

At the time that HTTPS is enabled, a new SSL key and self-signed certificate are automatically generated. with the following parameters:

Key length: 2048
Country-code: "US"
Org name: "Extreme Networks"
Common name: "*.extremenetworks.com"
State: "NC"
Locality: "Morrisville"
Org unit: "EXOS"

The **`show configuration`** command shows that HTTPS is enabled.

When upgrading to Switch Engine 32.7.1, HTTPS is not enabled when:

- HTTPS is disabled and a switch key by itself is present
- HTTPS is disabled and a certificate by itself is present
- HTTPS is disabled and both a switch key and a certificate is present
- HTTPS is enabled

Secure Shell 2

Secure Shell 2 (SSH2) is a feature of the ExtremeXOS software that allows you to encrypt session data between a network administrator using SSH2 client software and the switch or to send encrypted data from the switch to an SSH2 client on a remote system.

Configuration, image, public key, and policy files can be transferred to the switch using the Secure Copy Protocol 2 (SCP2).

The ExtremeXOS SSH2 switch application works with the following clients: Putty, SSH2 (version 2.x or later) from SSH Communication Security, and OpenSSH (version 2.5 or later).

SSH Server Overview

SSH server in ExtremeXOS device listens for incoming connections. The server uses various services in ExtremeXOS including AAA for authentication, Policy Manager for access control, Session Manager for session reporting, and EMS for logging.

Understanding SSH Server

A 2,048-bit RSA Host key for the SSH server on the ExtremeXOS device is generated. Once the key is generated, the SSH server is started and listens for connections. Password and public key authentication methods are supported.

The client's public keys are configurable through the CLI. A maximum of 100 keys can be created. A specific key can hold a maximum of 16 users, and a single user can be added to a maximum of 16 keys.

For password authentication, the configuration should be done in the AAA module, where an account must be created for the user, either as an admin or a user, and the corresponding password should be configured. The same will be used for validating the user, when exsshd (SSH server daemon/process) contacts AAA for authenticating the user.

Upon receiving a connection, it will be first validated with *ACL*/Policy, if configured. If the source-address from which the connection has been established, and is configured for a deny action in the acl, the connection is closed.

If the ACL permits the source-address of the client, the server proceeds with authentication steps. As part of authentication, the versions are exchanged between the client and the server. ExtremeXOS SSH server supports protocol version 2. If the client has a protocol version 1.99 or above, the same is accepted. Otherwise, the connection is rejected. Once the versions of both the ends are accepted, key exchange algorithms proposal of server and client are matched. Openssh-7.5p1 supports Diffie-Hellman group 1, 14, 16, and 18 as part of the key exchange algorithms (see Diffie-Hellman Overview on page 1674). The minimal supported Diffie-Hellman group is 14 by default in Default mode, since group 1 is considered weaker. After the key exchange algorithm is agreed upon, the ciphers are exchanged and agreed upon. Openssh-7.5p1 supports the following ciphers and MACs.

- **Ciphers**: aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se, aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
- **MACs**: hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-md5; hmac-sha1, hmac-sha2-256, hmac-sha2-512 hmac-sha1-96, hmac-md5-96

After the key exchanges, the local authentication methods that are supported (public key and password authentication) are tried first. If there is a matching key or password, the AAA authentication will be tried out. Else, the user will be considered to be unknown, and the session will be terminated.

Below are the steps that are involved between the client and the server while establishing the connection.

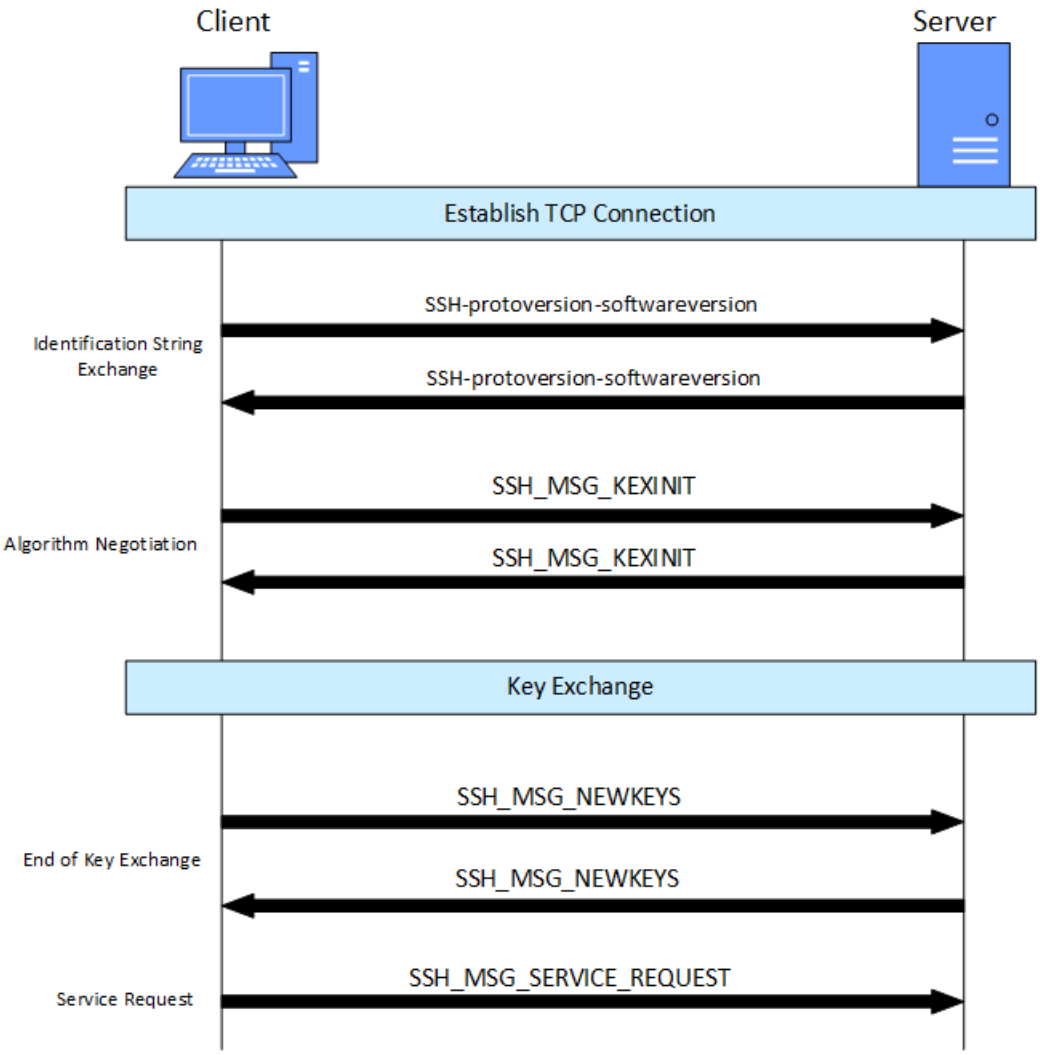


Figure 198: SSH Client /Server Connection Steps

Upon disabling SSH using the command `disable ssh2` , the server stops listening for new connections, but the existing connections continue to be active. When you enable the SSH server again, the key is not regenerated by default. Specific sessions or all the sessions can be closed through `clear session` command through CLI. The sessions are closed once the idle-timeout expires. By default the idle timeout is 3,600 seconds for SSH. The same can be configured through the CLI. If the session is left idle for more than the configured/default idle-timeout value in seconds, then the session is closed by the server. The idle timeout will be on per session basis. In other words, the idle timeout that is configured when the session is established, will be the one used for that particular session. If the user changes the idle timeout after the session is established, that will not be reflected on the existing sessions, but will be used for new sessions.

SSH Default Enabled Parameters During Installation

When you install Switch Engine for the first time, the following SSH parameters are enabled by default:

In Default, FIPS, and Secure mode:

- **Diffie-Hellman groups**: 14 (2,048 bits), 16 (4,096 bits), 18 (8192 bits)
- **Public key algorithms**: ssh-rsa, x509v3-sign-rsa, x509v3-sign-dss

In Default mode:

- **Ciphers**: aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
- **MACs**: hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1, hmac-sha2-256, hmac-sha2-512

Other OpenSSH 7.5p1 supported MACs and ciphers listed in Understanding SSH Server on page 1645 are disabled by default.

RSA and SHA256 for Host Key Algorithm

Version 32.5 adds support for two new host key algorithms: rsa-sha2-256 and rsa-sha2-512. While the default algorithm remains ssh-rsa, this SHA-1 algorithm is weak and not recommended. In version 32.5, you can use the CLI to select the host key algorithm from the list of three options.

During an upgrade to version 32.5, the ssh-rsa type host key present in the switch is used, but the following EMS log will be generated when the switch starts:

```
04/25/2023 08:19:25.67 <Noti:exsshd.CfgHostKeyAlgWeak> The configured host key
algorithm(s),
        ssh-rsa, is/are weaker than what is recommended.
```

The switch will continue to generate an ssh-rsa type key until you use the **configure ssh2 key algorithm** command. Once you use the command to make a selection, the new algorithm chosen will take effect when you run **disable/enable ssh2** or **sshd restart**, as displayed in the following example output:

```
# configure ssh2 key algorithm rsa-sha2-256
New key algorithm will be usable after disable and enable SSH or 'restart process exsshd'.

Warning: Legacy clients that do not support this algorithm will not connect with the
        switch's SSH server.
```

The following command configures the host key algorithm:

```
configure ssh2 key algorithm [ ssh-rsa | rsa-sha2-256 | rsa-sha2-512]
```

Use the **show ssh2** command to display current and configured algorithms.

Enabling SSH2 for Inbound Switch Access

You must enable SSH2 on the switch before you can connect to the switch using an external SSH2 client.

Enabling SSH2 involves two steps:

1. Generate or specify an authentication key for the SSH2 sessions (see Standard Key Authentication on page 1648 and User Key-Based Authentication on page 1648).
2. Enable SSH2 access by specifying a TCP port to be used for communication and specifying on which SSH2 is enabled (see Enabling SSH2 on page 1658).

3. If desired, specify a timeout period for a login attempts by using the command:
   `configure` **`ssh2 login-grace-timeout`** *`seconds`*

Standard Key Authentication

An authentication key must be generated before the switch can accept incoming SSH2 sessions. This can be done automatically by the switch, or you can enter a previously generated key.

- To have the key generated by the switch, use the following command:

  `configure ssh2 key`

  The key generation process can take up to one minute.

  For the switch to use the newly generated key, the exsshd process needs to be restarted using the `restart process [`**`class`** *`cname`* `|` *`name`* `{`**`msm`** *`slot`*`}]` command with "exsshd" for *`name`*.

- To use a key that has been previously created, use the following command:

  `configure ssh2 key {`**`pregenerated`**`}`

  The switch prompts you to enter the pregenerated key.

  Note
  The pregenerated key must be one that was generated by the switch. To get such a key, you can use the command `show ssh2 private-key` to display the key on the console. Copy the key to a text editor and remove the carriage return/line feeds. Finally, copy and paste the key into the command line. The key must be entered as one line.

The key generation process generates the SSH2 private host key. The SSH2 public host key is derived from the private host key and is automatically transmitted to the SSH2 client at the beginning of an SSH2 session.

User Key-Based Authentication

Public key authentication is an alternative method to password authentication that SSH uses to verify identity. You can generate a key pair consisting of a private key and a public key. The public key is used by the ExtremeXOS SSH server to authenticate the user.

In ExtremeXOS, user public keys are stored in the switch's configuration file; these keys are then associated (or bound) to a user.

The keys are configured on the switch in one of two ways:

- By copying the key to the switch using scp2/sftp2 with the switch acting as the server.
- By configuring the key using the CLI.

RSA and DSA encryption keys are both supported.

The public key can be loaded onto the switch using SCP or SFTP, where the switch is the server. The administrator can do this by using the SCP2 or SFTP2 client software

to connect to and copy the key file to the switch. The public key file must have the extension ssh; for example, id_dsa_2048.ssh. When the .ssh file is copied to the switch, the key is loaded into the memory. The loaded public keys are saved to the configuration file (*.cfg) when the save command is issued via the CLI.

The key name is derived from the file name. For example, the key name for the file id_dsa_2048.ssh will be id_dsa_2048. The file name of the key or the keyname is restricted to 32 characters in length.

The key is associated with a user either implicitly, by pre-pending the user name to the file or explicitly using the CLI.

In order for a key to be bound or associated to a user, the user must be known. In other words, that user must have an entry in the local database on the switch. Once the user is authenticated, the user's rights (read-only or read/write) are obtained from the database.

The key can be associated with a user by pre-pending the user name to the file name. For example, admin.id_dsa_2048.ssh.

If the user specified in the file name does not exist on the switch, the key is still accepted, but is not associated to any user. Once the user is added, the key can be associated with the user using the CLI. If the user name is not pre-pended to the file name, the key is accepted by the switch but is not associated with any user. The key can be then be associated with the user using the command `configure sshd2 user-key` *`key_name`* **`add user`** *`user_name`* .

You can also enter or paste the key using the command `configure ssh2 key {`**`pregenerated`**`}` using the pregenerated keyword. There cannot be any carriage returns or new lines in the key. For more information, see the `configure ssh2 key {`**`pregenerated`**`}` command in the Switch Engine v33.6.1 Command References

The host and user public keys can be written to a file in the `config` directory using the `create sshd2 key-file {`**`host-key`**` | `**`user-key`**`}` *`key_name`* command. This enables the administrator to copy the public key to an outside server.

Public-Key Infrastructure (PKI) in Secure Shell (SSH) Overview

The major disadvantage with user-key-based authentication is scalability. As the number of users increases, a greater number of keys must be copied and stored in the switch. This problem can be solved with the Public-Key Infrastructure (PKI). Moreover, PKI has its own advantages of added security, certificate revocation checking, avoiding manual mapping of keys with users, etc.

For the details about configuring PKI, see Using Public-Key Infrastructure (PKI) in Your Network on page 1680.

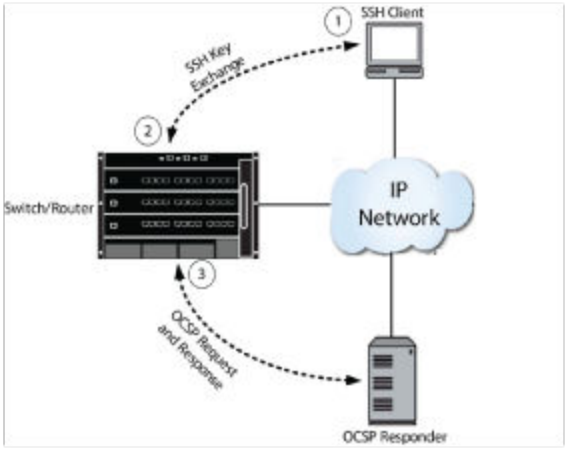


Figure 199: Public-Key Infrastructure Login Flow Overview

In Figure 199, Callout 1 is the initial series of message exchanges initiated by the Secure Shell (SSH) client. The ExtremeXOS device is providing the SSH client with the list of supported authentication methods, one of which is public key. The SSH client responds with its public key certificate.

At Callout 2, the ExtremeXOS device verifies if the extended key usage of client certificate contains 'client authentication'. If not, the SSH PKI connection is not established. Next, the ExtremeXOS device extracts the Common-Name field from the public key certificate and validates it against the local-accounts present/configured in the switch. If there is no matching accounts/user-name, then the SSH PKI connection is not established. Next, it checks to ensure the certificate signature from SSH client matches a trusted certificate authority's certificate present on the ExtremeXOS device.

At Callout 3, the ExtremeXOS device sends an Online Certificate Status Protocol (OCSP) request to the OCSP responder to check the validity of the Clients X.509 certificate. If the OCSP responder determines that the certificate has not been revoked by the certificate authority, the server sends back a GOOD response. After the preceding steps are completed, then the user is logged in automatically using SSH with X509, which can be seen using the command `show session {{detail} {sessID}} {history}` with the output shown as follows (the Auth field has a value of x509v3, which indicates that this is a SSH login using PKI):

```
# show session
                                                                    CLI
    #       Login Time               User     Type    Auth    Auth Location
================================================================================
*5          Tue Oct 18 12:24:12 2016 samar .. ssh2    x509v3 dis  10.127.3.143
```

Limitations

- Certificate-based authentication is supported only for ExtremeXOS Secure Shell (SSH) server, not for ExtremeXOS SSH client.
- Revocation check is done only for the SSH client-end certificate using OCSP only at the time of login. There are no periodic revocation checks.

- The SSH client certificate must have client authentication purpose in the extendedKeyUsage field.
- Username must be present in CommonName (CN) in the subject of the certificate. The login username and this CN must match for access to be granted.
- Supports only RSA, DSA-based SSH client certificates.
- Beginning with Release 32.2, if X.509v3 authentication fails due to an invalid key or certificate, then the connection request will be rejected. The next authentication method will not be attempted.

Example: Configuring PKI for SSH Secure Login Using X509v3 Certificates

The following is an example on how to use the ExtremeXOS PKI for SSH secure login using X509v3 certificates.

For example purposes, the following has been considered:

- OpenSSL is used to generate the trusted CA certificates and SSH client certificates.
- CommonIssuerModel is used for OCSP.
- OpenSSL's OCSP utility is used.
- Roumenpetrov's "PKIX-SSH" withX.509 v3 certificate support is used as an SSH client.

See the following topics for an example:

- Generating Certificates Using OpenSSL on page 1651
- Downloading the Trusted Certificate to the ExtremeXOS PKI and Configuring the User Accounts on page 1654
- Using OpenSSL OCSP Utility on page 1654
- Using Roumenpetrov's "PKIX-SSH" SSH Client for Logging on page 1654
- The openssl.cnf file Used in This Example on page 1655

Generating Certificates Using OpenSSL

The following steps should be followed on a Linux server with OpenSSL installed to the generate the X509v3 certificates.

```
$ mkdir certs crl newcerts serial private
$ touch index.txt
$ echo "01" > serial



$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -subj "/C=US/ST=NC/L=Raleigh/O=Extr/
OU=Exos/CN=CA-EXOS/emailAddress=ca-exos@extremenetworks.com" -keyout exosCAkey.pem -out
exosCAcert.crt
Generating a 2048 bit RSA private key
.+++
...............................................+++
writing new private key to 'exosCAkey.pem'
-----

$ openssl x509 -in exosCAcert.crt -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 17387834014905383023 (0xf14dfee96fe3b86f)
    Signature Algorithm: sha256WithRSAEncryption
```

```
        Issuer: C=US, ST=NC, L=Raleigh, O=Extr, OU=Exos, CN=CA-EXOS/emailAddress=ca-
exos@extremenetworks.com
        Validity
            Not Before: Oct 20 07:12:47 2016 GMT
            Not After : Oct 20 07:12:47 2017 GMT
        Subject: C=US, ST=NC, L=Raleigh, O=Extr, OU=Exos, CN=CA-EXOS/emailAddress=ca-
exos@extremenetworks.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c3:6a:11:b0:13:ad:a0:a4:fe:e7:5a:60:26:e4:
                    47:7c:4e:1b:66:5f:ae:19:60:58:38:e9:31:09:07:
                    4b:5d:a0:80:a6:fb:ff:21:1e:bc:cd:f9:f0:0a:1f:
                    e4:ec:06:ab:1f:37:2d:64:d6:13:77:47:6b:e8:81:
                    d5:da:0a:e0:96:09:b4:9c:bd:5f:6a:dc:5e:04:5f:
                    5f:8e:79:a8:1c:6e:ba:29:63:8c:d1:f8:c6:53:40:
                    66:dc:21:0c:f4:38:c4:81:57:e7:8e:b1:d9:c6:e2:
                    cb:c6:b2:80:f4:fb:da:2f:e0:45:3f:15:5c:b4:92:
                    90:0b:6c:18:88:c4:45:67:a0:93:d1:88:9d:bb:51:
                    d3:d5:1e:55:4a:31:5d:a6:d3:3f:53:81:82:e2:9d:
                    27:b1:34:4e:06:2e:20:52:fd:3d:3a:5d:c3:9d:b1:
                    63:02:95:37:67:3f:f8:e4:8b:31:c7:c2:fe:08:36:
                    aa:95:92:78:2f:88:09:83:8b:94:87:23:f6:7d:ed:
                    20:75:87:79:14:b7:0e:1d:81:c8:93:e0:11:6c:e4:
                    97:3b:a9:f6:1f:34:c3:6f:94:a8:ec:9e:4f:6d:02:
                    37:2e:34:c1:01:24:6e:10:ca:7d:c9:c9:f2:12:5c:
                    ef:e1:b2:d1:67:bb:f2:15:94:e9:4b:70:11:a7:35:
                    fd:69
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                FB:04:BC:52:1B:92:1C:80:8D:81:E0:D7:3E:16:91:59:D0:90:19:D8
            X509v3 Authority Key Identifier:
                keyid:FB:04:BC:52:1B:92:1C:80:8D:81:E0:D7:3E:16:91:59:D0:90:19:D8

            X509v3 Basic Constraints:
                CA:TRUE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment, Certificate Sign
    Signature Algorithm: sha256WithRSAEncryption
         36:85:1a:96:49:8f:9b:51:7d:9e:c2:cd:31:72:0a:58:08:b4:
         58:6b:8b:22:37:71:ed:b6:ec:96:98:3c:8f:6b:03:2b:f9:4d:
         c7:81:40:97:10:df:d3:24:79:71:86:2f:f4:21:48:64:a6:8e:
         50:09:40:84:de:3b:55:cf:f5:bc:a3:21:00:92:19:08:d1:cf:
         7f:df:e9:01:07:f0:2e:0e:b5:3d:71:fd:75:bf:4c:ae:bc:01:
         65:78:31:48:21:af:b6:fd:56:bf:a0:f7:33:62:e2:1f:64:79:
         01:b0:70:bf:9b:57:e6:75:ec:81:4f:92:3c:70:27:b7:17:3d:
         59:b4:f4:c5:09:55:a2:6e:77:75:26:e4:6f:63:9b:60:3f:0f:
         a0:b0:01:cf:de:5d:83:b7:dd:75:68:55:c0:69:b1:49:17:1f:
         23:16:8a:02:ae:ad:3f:5c:ef:98:a9:f7:58:0b:73:58:51:fe:
         0f:20:24:e3:29:c7:ea:36:66:b1:f1:d6:bb:79:ce:87:e4:22:
         83:9e:63:cc:2a:2f:69:45:01:04:ff:d0:a3:7f:46:ef:ab:da:
         ca:06:ad:ee:40:1b:15:39:f8:db:b6:47:45:11:83:8c:e4:54:
         59:75:7e:41:ce:54:fe:04:83:4e:30:db:6a:d6:23:4d:66:17:
         44:dd:7f:49
-----BEGIN CERTIFICATE-----
MIID8jCCAtqgAwIBAgIJAPFN/ulv47hvMA0GCSqGSIb3DQEBCwUAMIGIMQswCQYD
VQQGEwJVUzELMAkGA1UECAwCTkMxEDAOBgNVBAcMB1JhbGVpZ2gxDTALBgNVBAoM
BEV4dHIxDTALBgNVBAsMBEV4b3MxEDAOBgNVBAMMB0NBLUVYT1MxKjAoBgkqhkiG
9w0BCQEWG2NhLWV4b3NAZXh0cmVtZW5ldHdvcmtzLmNvbTAeFw0xNjEwMjAwNzEy
NDdaFw0xNzEwMjAwNzEyNDdaMIGIMQswCQYDVQQGEwJVUzELMAkGA1UECAwCTkMx
EDAOBgNVBAcMB1JhbGVpZ2gxDTALBgNVBAoMBEV4dHIxDTALBgNVBAsMBEV4b3Mx
EDAOBgNVBAMMB0NBLUVYT1MxKjAoBgkqhkiG9w0BCQEWG2NhLWV4b3NAZXh0cmVt
```

```
ZW5ldHdvcmtzLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMNq
EbATraCk/udaYCbkR3xOG2ZfrhlgWDjpMQkHS12ggKb7/yEevM358Aof5OwGqx83
LWTWE3dHa+iB1doK4JYJtJy9X2rcXgRfX455qBxuuiljjNH4xlNAZtwhDPQ4xIFX
546x2cbiy8aygPT72i/gRT8VXLSSkAtsGIjERWegk9GInbtR09UeVUoxXabTP1OB
guKdJ7E0TgYuIFL9PTpdw52xYwKVN2c/+OSLMcfC/gg2qpWSeC+ICYOLlIcj9n3t
IHWHeRS3Dh2ByJPgEWzklzup9h80w2+UqOyeT20CNy40wQEkbhDKfcnJ8hJc7+Gy
0We78hWU6UtwEac1/WkCAwEAAaNdMFswHQYDVR0OBBYEFPsEvFIbkhyAjYHg1z4W
kVnQkBnYMB8GA1UdIwQYMBaAFPsEvFIbkhyAjYHg1z4WkVnQkBnYMAwGA1UdEwQF
MAMBAf8wCwYDVR0PBAQDAgLkMA0GCSqGSIb3DQEBCwUAA4IBAQA2hRqWSY+bUX2e
ws0xcgpYCLRYa4siN3HttuyWmDyPawMr+U3HgUCXEN/TJHlxhi/0IUhkpo5QCUCE
3jtVz/W8oyEAkhkI0c9/3+kBB/AuDrU9cf11v0yuvAFleDFIIa+2/Va/oPczYuIf
ZHkBsHC/m1fmdeyBT5I8cCe3Fz1ZtPTFCVWibnd1JuRvY5tgPw+gsAHP3l2Dt911
aFXAabFJFx8jFooCrq0/XO+YqfdYC3NYUf4PICTjKcfqNmax8da7ec6H5CKDnmPM
Ki9pRQEE/9Cjf0bvq9rKBq3uQBsVOfjbtkdFEYOM5FRZdX5BzlT+BINOMNtq1iNN
ZhdE3X9J
-----END CERTIFICATE-----
```

The next steps are to create a X509v3 certificate for a user that is signed by the above generated CA certificate "exosCAcert.crt".

In the following command, a user certificate signing request with RSA 2048 bit key, commonName as "exos-admin" is generated:

```
$ openssl req -nodes -days 365 -newkey rsa:2048 -new -subj "/C=US/ST=NC/L=Raleigh/
O=Extr/OU=Exos/CN=exos-admin/emailAddress=exos-admin@extremenetworks.com" -keyout exos-
admin-key.pem -out exos-admin-req.csr
Generating a 2048 bit RSA private key
.........................................+++
.....................................+++
writing new private key to 'exos-admin-key.pem'
```

In the following command, the user certificate-signing-request that was generated in the preceding command is being signed using the CA certificate "exosCAcert.crt":

```
$ openssl ca -config openssl.cnf -extensions usr_cert -days 365 -keyfile exosCAkey.pem
-cert exosCAcert.crt -in exos-admin-req.csr -out exos-admin-cert.crt
Using configuration from openssl_A.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Oct 20 07:39:04 2016 GMT
            Not After : Oct 20 07:39:04 2017 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NC
            organizationName          = Extr
            organizationalUnitName    = Exos
            commonName                = exos-admin
            emailAddress              = exos-admin@extremenetworks.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                EE:A8:8F:6D:00:CA:93:57:22:E6:1F:DF:43:B4:91:E9:DE:B8:9F:D3
            X509v3 Authority Key Identifier:
                keyid:FB:04:BC:52:1B:92:1C:80:8D:81:E0:D7:3E:16:91:59:D0:90:19:D8
                DirName:/C=US/ST=NC/L=Raleigh/O=Extr/OU=Exos/CN=CA-EXOS/emailAddress=ca-
exos@extremenetworks.com
                serial:F1:4D:FE:E9:6F:E3:B8:6F

            X509v3 Key Usage: critical
```

```
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, E-mail Protection
            Authority Information Access:
                OCSP - URI:http://ocspserver.extremenetworks.com:2561

Certificate is to be certified until Oct 20 07:39:04 2017 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Downloading the Trusted Certificate to the ExtremeXOS PKI and Configuring the User Accounts

The previously generated trusted CA certificate needs to be downloaded from a TFTP server using the following command. "10.127.3.236" is the TFTP server IP address:

```
DUT1_switch_model.3 # download ssl 10.127.3.236 certificate trusted-ca /certs/
exosCAcert.crt
 DUT1_switch_model.4 #
 DUT1_switch_model.4 # sh ssl trusted-ca exosCAcert.crt
================ exosCAcert.crt ================
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f1:4d:fe:e9:6f:e3:b8:6f
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=NC, L=Raleigh, O=Extr, OU=Exos, CN=CA-EXOS/emailAddress=ca-
exos@extremenetworks.com
        Validity
            Not Before: Oct 20 07:12:47 2016 GMT
            Not After : Oct 20 07:12:47 2017 GMT
        Subject: C=US, ST=NC, L=Raleigh, O=Extr, OU=Exos, CN=CA-EXOS/emailAddress=ca-
exos@extremenetworks.com
```

The user needs to be created on the ExtremeXOS device with appropriate privileges. The name of the account would be the value present in the “commonName” field of the user-certificate that was generated (that is: exos-admin-cert.crt):

```
DUT1_switch_model.1 # create account admin exos-admin exospassword
```

Using OpenSSL OCSP Utility

As mentioned previously, OpenSSL’s OCSP utility is used for example purposes, and the example uses the “Common Issuer Model”.

The following command should be entered in the Linux Server for OCSP Responder:

```
$openssl ocsp -port 2561 -text -index index.txt -CA exosCAcert.crt
-rkey exosCAkey.pem -rsigner exosCAcert.crt
```

Using Roumenpetrov’s “PKIX-SSH” SSH Client for Logging

To use the user-certificate for logging into the ExtremeXOS SSH server, you need to concatenate the key file and certificate file as a single file, and also ensure that the file does not have a very open permission level (that is, accessible by others).

```
$ cat exos-admin-key.pem exos-admin-cert.crt > exosadmincert.crt
$ chmod 0600 exosadmincert.crt
```

```
$ ssh -i exosadmincert.crt exos-admin@10.127.6.57

ExtremeXOS
Copyright (C) 1996-2016 Extreme Networks. All rights reserved.
This product is protected by one or more US patents listed at http://
www.extremenetworks.com/patents along with their foreign counterparts.
================================================================================

Press the <tab> or '?' key at any time for completions.


DUT1_switch_model.1 # show session
                                                                  CLI
    #       Login Time               User     Type    Auth    Auth Location
================================================================================
*7          Thu Oct 20 20:25:37 2016 exos- .. ssh2    x509v3 dis  10.127.3.143

DUT1_switch_model.2 # show session detail

ID                  :7
Start Time          :Thu Oct 20 20:25:37 2016
User Name           :exos-admin
Type                :ssh2
Authentication      :x509v3
Cli Authentication  :disabled
Location            :10.127.3.143
Node                :local

DUT1_switch_model.3 # sh log
10/20/2016 20:25:37.17 <Info:AAA.LogSsh> Msg from Master : Did x509v3 authentication for
user exos-admin (10.127.3.143)
10/20/2016 20:25:37.17 <Info:AAA.LogSsh> Msg from Master : Login passed for user exos-
admin through ssh (10.127.3.143)
```

The openssl.cnf file Used in This Example

```
[ ca ]
# `man ca`
default_ca = CA_default

[ CA_default ]
# Directory and file locations.
dir               = /root
certs             = $dir/certs
crl_dir           = $dir/crl
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand
unique_subject = no

# The root key and root certificate.
private_key       = $dir/private/ca.key.pem
certificate       = $dir/certs/ca.cert.pem

# For certificate revocation lists.
crlnumber         = $dir/crlnumber
crl               = $dir/crl/ca.crl.pem
crl_extensions    = crl_ext
default_crl_days  = 30

# SHA-1 is deprecated, so use SHA-2 instead.
```

```
default_md        = sha256

name_opt          = ca_default
cert_opt          = ca_default
default_days      = 375
preserve          = no
policy            = policy_strict

[ policy_strict ]
# The root CA should only sign intermediate certificates that match.
# See the POLICY FORMAT section of `man ca`.
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ policy_loose ]
# Allow the intermediate CA to sign a more diverse range of certificates.
# See the POLICY FORMAT section of the `ca` man page.
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = optional
emailAddress            = optional

[ req ]
# Options for the `req` tool (`man req`).
default_bits        = 2048
distinguished_name  = req_distinguished_name
string_mask         = utf8only

# SHA-1 is deprecated, so use SHA-2 instead.
default_md          = sha256

# Extension to add when the -x509 option is used.
x509_extensions     = v3_ca

[ req_distinguished_name ]
# See <https://en.wikipedia.org/wiki/Certificate_signing_request>.
countryName                     = Country Name (2 letter code)
stateOrProvinceName             = State or Province Name
localityName                    = Locality Name
0.organizationName              = Organization Name
organizationalUnitName          = Organizational Unit Name
commonName                      = Common Name
emailAddress                    = Email Address

# Optionally, specify some defaults.
countryName_default             = GB
stateOrProvinceName_default     = England
localityName_default            =
0.organizationName_default      = Alice Ltd
organizationalUnitName_default  =
emailAddress_default            =

[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
```

```
keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ v3_intermediate_ca ]
# Extensions for a typical intermediate CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
#nsCertType = client, email
#nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection
authorityInfoAccess = OCSP;URI:http://ocspserver.extremenetworks.com:2561

[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth

[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier=keyid:always

[ ocsp ]
# Extension for OCSP signing certificates (`man ocsp`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning
```

SSH X.509v3 Authentication Using RADIUS

Version 32.2 introduces support for SSH X.509v3 Authentication Using RADIUS. When enabled, the SSH server requests a password from the client during X.509v3 key-certificate authentication. The user name and password received is then sent to Authentication, Authorization, and Accounting (AAA) for authentication. AAA sends these credentials to the RADIUS server for authentication. If the RADIUS server is not configured or is not rechable, then AAA will validate the user name and password through local authentication. You can use the following command to enable or disable the RADIUS password authentication feature:

```
configure ssh2 x509v3 radius-password-auth [on | off]
```

For authentication, you can use Principal Name as the user name in the client's certificate. The ‘principalName’ (OID: 1.3.18.0.2.4.318) is displayed below the ‘otherName’ identifier in the ‘SubjectAltName’ (SAN) extension of the X509v3 key-certificate.

The following is an example of openssl.cnf to generate X509v3 certificate with SAN extension having ‘prinicipalName’:

```
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
#nsCertType = client, email
#nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection
authorityInfoAccess = OCSP;URI:http://ocspserver.extremenetworks.com:2561
subjectAltName = otherName:1.3.18.0.2.4.318;UTF8:exos-admin@sustaining.com
```

Overwrite

The `configure ssh2` **`x509v3 username overwrite`** [**`on`** | **`off`**] command is used to enable or disable using Principal Name as the username for authentication. If 'principalName' is not found, then Common Name in the certificate will be used as the username. This command can only be used if `radius-password-auth` is enabled.

Strip Domain

The `configure ssh2` **`x509v3 username strip-domain`** [**`on`** | **`off`**] command is used to specify using the 'principalName' without its domain name. This command can only be used if `radius-password-auth` and `overwrite` are enabled.

Use Domain

The `configure ssh2` **`x509v3 username use-domain`** [*`domain_name`* | **`none`**] command is used to add a domain name to the username for authentication. This command can only be used if `radius-password-auth`, `overwrite`, and `strip-domain` are enabled.

Enabling SSH2

- To enable SSH2, use the following command:

  `enable ssh2` {**`access-profile`** [*`access_profile`* | **`none`**]} {**`port`** *`tcp_port_number`*} {**`vr`** [*`vr_name`* | **`all`** | **`default`**]}

  You can also specify a TCP port number to be used for SSH2 communication. The TCP port number is 22 by default. The switch accepts IPv6 connections.

  Before you initiate a session from an SSH2 client, ensure that the client is configured for any non-default access list or TCP port information that you have configured on the switch. After these tasks are accomplished, you may establish an SSH2-encrypted session with the switch. Clients must have a valid user name and password on the switch in order to log in to the switch after the SSH2 session has been established.

  Up to eight active SSH2 sessions can run on the switch concurrently. If you enable the idle timer using the `enable cli idle-timeout` command, the SSH2 connection times out after 20 minutes of inactivity by default. If you disable the idle timer using the `disable cli idle-timeout` command, the SSH2 connection times

out after 60 minutes of inactivity by default. This timeout value can be modified using the `configure ssh2 idletimeout minutes` command wherein *minutes* can be from 1 to 240 ". For more information please refer to the command help for "configure ssh2".

After SSH2 has enabled, TCP port 22 is available on all virtual routers by default .

General technical information is also available at http://www.openssh.com/specs.html.

Setting SSH2 Rekeying Interval

Data transmission between server and client in each SSH2 session is encrypted using session keys.

Session keys are generated after negotiation between server and client using Diffie-Hellman algorithm (see Diffie-Hellman Overview on page 1674). Cryptanalysis experts advise that it is unsafe to use the same session key to encrypt data over long periods of time. With enough captured data you could analyze the traffic and compromise the key, so it is advisable to keep changing the session keys after a certain interval.

You can configure the SSHv2 session rekeying interval by specifying a time interval and/or a data limit. After the configured time interval, the SSH server forces the client to perform a key negotiation for a new session key. This new key is used for SSH communication until the next rekeying. After configuring the SSH session rekeying time interval, configured SSH idle-timeout is disabled, so the idle-timeout occurs at the globally configured idle-timeout, instead of SSH configured idle-timeout.

Default value for data limit varies depending upon the encryption algorithm being used. For algorithms with a block size of 128-bits, data-limit is 4GB; for algorithms with a block size of 64-bits, the data limit is 1GB. This is the limitation by OpenSSH. For example, if the communication is through 3-DES algorithm and the data limit is configured at 4GB, rekeying occurs after 1GB of data is transferred. If the communication is using AES algorithm with same configuration, rekeying occurs after 4GB data is transferred. This issue does not occur if you limit the range to 1GB, instead of 4GB for the data limit.

To set the rekey interval, use the following command:

```
configure ssh2 rekey [time-interval [time_interval |none] |data-limit
[data_size |default]]
```

To view your selections for rekeying, use the following commands:

```
show ssh2
```

or

```
show management
```

Enabling/Disabling DSA and X509v3 Public Key Algorithms

You can selectively enable and disable DSA/RSA X509 public key-based algorithms: ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss.

To enable DSA/RSA X509 public key-based algorithms, use the following command:

```
configure ssh2 enable {pk-alg [pkalg_name | all]}
```

To disable DSA/RSA X509 public key-based algorithms, use the following command:

```
configure ssh2 disable {pk-alg [pkalg_name | all]}
```

To view which public key-based algorithms are currently enabled, use the following command:

```
show ssh2
```

Viewing SSH2 Information

- To view the status of SSH2 sessions on the switch, use the following command:

  ```
  show ssh2
  ```

  The `show switch management` command displays information about the switch including the enable/disable state for SSH2 sessions and whether a valid key is present.

Using ACLs to Control SSH2 Access

You can restrict SSH2 access by creating and implementing an *ACL* policy.

You configure an ACL policy to permit or deny a specific list of IP addresses and subnet masks for the SSH2 port.

The two methods to load ACL policies to the switch are:

- Use the `edit policy` command to launch a VI-like editor on the switch. You can create the policy directly on the switch.
- Use the `tftp` command to transfer a policy that you created using a text editor on another system to the switch.

For more information about creating and implementing ACLs and policies, see Security and ACLs on page 341.

Sample SSH2 Policies

The following are sample policies that you can apply to restrict SSH2 access.

In the following example, named MyAccessProfile.pol, the switch permits connections from the subnet 10.203.133.0/24 and denies connections from all other addresses:

```
MyAccessProfile.pol
Entry  AllowTheseSubnets {
if  {
source-address 10.203.133.0 /24;
}
Then
{
permit;
}
}
```

In the following example, named MyAccessProfile.pol, the switch permits connections from the subnets 10.203.133.0/24 or 10.203.135.0/24 and denies connections from all other addresses:

```
MyAccessProfile.pol
Entry  AllowTheseSubnets {
if match any {
source-address 10.203.133.0 /24;
source-address 10.203.135.0 /24;
}
Then
{
permit;
}
}
```

In the following example, named MyAccessProfile_2.pol, the switch does not permit connections from the subnet 10.203.133.0/24 but accepts connections from all other addresses:

```
MyAccessProfile_2.pol
Entry dontAllowTheseSubnets {
if {
source-address 10.203.133.0 /24;
}
Then
{
deny;
}
}
Entry  AllowTheRest {
If  {
; #none specified
}
Then
{
permit;
}
}
```

In the following example, named MyAccessProfile_2.pol, the switch does not permit connections from the subnets 10.203.133.0/24 or 10.203.135.0/24 but accepts connections from all other addresses:

```
MyAccessProfile_2.pol
Entry dontAllowTheseSubnets {
if match any {
source-address 10.203.133.0 /24;
source-address 10.203.135.0 /24
}
Then
{
deny;
}
}
Entry  AllowTheRest {
If  {
; #none specified
}
Then
{
permit;
```

```
}
}
```

Configuring SSH2 to Use ACL Policies

This section assumes that you have already loaded the policy on the switch. For more information about creating and implementing *ACL*s and policies, see Security and ACLs.

- To configure SSH2 to use an ACL policy to restrict access, use the following command:

  `enable ssh2 {`**`access-profile`** `[`*`access_profile`* `|` **`none`**`]} {`**`port`** *`tcp_port_number`*`} {`**`vr`** `[`*`vr_name`* `|` **`all`** `|` **`default`**`]}`

  Use the none option to remove a previously configured ACL.

  In the ACL policy file for SSH2, the source-address field is the only supported match condition. Any other match conditions are ignored.

Using SCP2 from an External SSH2 Client

In ExtremeXOS, the SCP2 protocol is supported for transferring configuration, image and public key policy files to the switch from the SCP2 client. The user must have administrator-level access to the switch. The switch can be specified by its switch name or IP address. ExtremeXOS only allows SCP2 to transfer to the switch files named as follows:

- *.cfg—ExtremeXOS configuration files
- *.pol—ExtremeXOS policy files
- *.xos—ExtremeXOS core image files
- *.xmod—ExtremeXOS modular package files
- *.ssh—Public key files

In the following examples, you are using a Linux system to move files to and from the switch at 192.168.0.120, using the switch administrator account admin. You are logged into your Linux system as user.

- To transfer the primary configuration file from the switch to your current Linux directory using SCP2, use the following command:

  ```
  [user@linux-server]# scp2 admin@192.168.0.120:primary.cfg primary.cfg
  ```

- To copy the policy filename test.pol from your Linux system to the switch, use the following command:

  ```
  [user@linux-server]# scp2 test.pol admin@192.168.0.120:test.pol
  ```

- To copy the image file test.xos from your Linux system to the switch, use the following command:

  ```
  [user@linux-server]# scp2 test.xos admin@192.168.0.120:test.xos
  ```

  Now you can use the command install image test.xos to install the image in the switch.

- To copy the SSH image file test.xmod from your Linux system to the switch, use the following command:

  ```
  [user@linux-server]# scp2 test.xmod admin@192.168.0.120:test.xmod
  ```

Now you can use the command install image test.xmod to install the image in the switch.

- To load the public key id_rsa.pub from your Linux system to the switch, use the following command:

```
[user@linux-server]# scp2 id_rsa.pub admin@192.168.0.120:test.ssh
```

This command loads the key into memory, which can be viewed with the command `show sshd2 user-key`.

Understanding the SSH2 Client Functions on the Switch

An Extreme Networks switch can function as an SSH2 client. This means you can connect from the switch to a remote device running an SSH2 server and send commands to that device. You can also use SCP2 to transfer files to and from the remote device.

Note

ExtremeXOS 15.7.1 upgraded from openssh-3.9p1 to openssh-6.5p1. ExtremeXOS 21.1 adds the openssl-fips-ecp-2.0.9 open source library. ExtremeXOS 22.5 upgraded from openssh-6.5p1 to openssh-7.5p1. ExtremeXOS 30.7 upgraded from openssh-7.5p1 to openssh-8.1p1

Support for following ciphers and macs are removed in ExtremeXOS 30.7, since these are not supported in openssh 8.1p1:

- Ciphers: blowfish-cbc, cast128-cbc, arcfour, arcfour256, arcfour128
- MACs: hmac-ripemd160-etm@openssh.com, hmac-ripemd160, hmac-ripemd160@openssh.com

Note

When upgrading to ExtremeXOS 30.7, if unsupported ciphers/MACs are:

- Enabled in the saved configuration, the configuration is ignored during configuration load. The dirty bit is set and an error message appears:. `<Erro:exsshd.LoadCfgCipherUnsuprt>` `<Erro:exsshd.LoadCfgMACUnsuprt>`.
- Disabled (by default or intentional) in the saved configuration, the configuration is ignored silently during configuration load. The dirty bit is set and an error messages do not appear.

You do not need to enable SSH2 or generate an authentication key to use the SSH2 and SCP2 commands from the ExtremeXOS CLI.

Note

User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

- To send commands to a remote system using SSH2, use the following command:

```
ssh2 {cipher [cipher] {mac mac} {compression [on | off]} {port port}
{user username} {vr vr_name} user@host {remote_command}
```

The remote commands can be any command acceptable by the remote system. You can specify the login user name as a separate argument or as part of the user@host specification. If the login user name for the remote system is the same as your user name on the switch, you can omit the username parameter entirely.

For example, to obtain a directory listing from a remote Linux system with IP address 10.10.0.2 using SSH2, enter the following command:`ssh2 admin@10.10.0.2 ls`

- To initiate a file copy from a remote system to the switch using SCP2, use the following command:

  `scp2 {`**`cipher`** *`cipher`*`} {`**`mac`** *`mac`*`} {`**`compression`** `[`**`on`** `|` **`off`**`]} {`**`port`** *`port`*`} {`**`vr`** *`vr_name`*`} [` *`user@host:file local-file | local-file user@host:file`* `]`

  For example, to copy the configuration file test.cfg on host system1 to the switch, enter the following command:

  ```
  scp2 admin@system1:test.cfg localtest.cfg
  ```

- To initiate a file copy to a remote system from the switch using SCP2, use the following command:

  `scp2 {`**`cipher`** *`cipher`*`} {`**`mac`** *`mac`*`} {`**`compression`** `[`**`on`** `|` **`off`**`]} {`**`port`** *`port`*`} {`**`vr`** *`vr_name`*`} [` *`user@host:file local-file | local-file user@host:file`* `]`

  For example, to copy the configuration file engineering.cfg from the switch to host system1, enter the following command:

  ```
  scp2 engineering.cfg admin@system1:engineering.cfg
  ```

SSH/SCP Client Upgrade Limitations

- Only password-based authentication is supported for SSH/SCP client.
- SCP client will not support upload of Image or BootROM files (i.e, *.xos, *.xmod, and *.xtr).
- SCP client will not support download of Image or BootRom files (like *.xos, *.xmod, and *.xtr) from ExtremeXOS SCP server.
- Only supports transfer of files such as *.cfg, *.xsf, *.py, *.pol and *.ssh files to/from the switch.
- Keyboard interactive authentication will not be supported.
- Host key algorithms are not configurable.

Using SFTP from an External SSH2 Client

The SFTP protocol is supported for transferring configuration, and policy files to the switch from the SFTP client. You must have administrator-level access to the switch. The switch can be specified by its switch name or IP address.

ExtremeXOS requires that SFTP transfer to the switch files named as follows:

- *.cfg—ExtremeXOS configuration files
- *.pol—ExtremeXOS policy files
- *.xos—ExtremeXOS core image file
- *.xmod—ExtremeXOS modular package file
- *.ssh—Public key files

In the following examples, you are using a Linux system to move files to and from the switch at 192.168.0.120, using the switch administrator account admin. You are logged into your Linux system as account user.

- To transfer the primary configuration file from the switch to your current Linux directory using SCP2, use the following command:

```
user@linux-server]# sftp admin@192.168.0.120
password: <Enter password>
sftp> put primary.cfg
```

- To copy the policy filename test.pol from your Linux system to the switch, use the following command:

```
user@linux-server]# sftp admin@192.168.0.120
password: <Enter password>
sftp> put test.pol
```

- To copy the image file `image.xos` from your Linux system to the switch, use the following command:

```
user@linux-server]# sftp admin@192.168.0.120
password: <Enter password>
sftp> put image.xos
```

- To load the public key id_rsa.pub from your Linux system to the switch, use the following command:

```
user@linux-server]# sftp admin@192.168.0.120
password: <Enter password>
sftp> put id_rsa.pub id_rsa.ssh
```

  For image file transfers, only one image file at a time can be available for installation. If you need to install an additonal image (xmod):

  a. Transfer `image.xos` into the switch using scp/sftp
  b. Install the `image.xos` image using the "install image" command
  c. Transfer the xmod into the switch using scp/sftp
  d. Install the xmod modular package file using `install image [`**`inactive`** `|` *`filename`* `|` *`local-file`*`] {`*`partition`*`} {`**`slot`** *`slotid`*`}{`**`reboot`**`}` command.

- For image file transfers using SFTP or SCP (with the switch acting as the server), once the image is copied to the switch, validation of image is done by the switch, as indicated by the following log message:

  ```
  <Info:AAA.LogSsh> Validating Image file, this could take approximately
  30 seconds..image.xos
  ```

  You must receive the following log message before you can proceed with the installation of the image:

  ```
  <Info:AAA.LogSsh> Image file successfully validated
  ```

  In stacking switches, you must receive the following log message from all slots before proceeding with the installation.

  For example, in a four-switch stack, the installation can be proceed only after the following log messages are received:

  ```
  04/19/2007 17:41:09.71 <Info:AAA.LogSsh> Slot-1: Sent file "image.xos" info to backup
  04/19/2007 17:41:09.71 <Info:AAA.LogSsh> Slot-1: Sent file "image.xos" info to standby
  slot 3
  ```

```
04/19/2007 17:41:09.71 <Info:AAA.LogSsh> Slot-1: Sent file "image.xos" info to standby
slot 4
```

Disabling Unapproved Crypto Algorithms

ExtremeXOS provides a way to administratively disable unapproved crypto algorithms. This allows you to enable and disable ciphers and MAC algorithms to be used by the SSH server.

Disabling Unapproved Crypto Algorithms

SSH works on three security modes default-mode, FIPS-mode, and Secure-Mode (see Security Mode Overview on page 1533). In each mode, you can configure the required ciphers and MACs from a list using the CLI. Different modes support a different list of ciphers. By default, all available ciphers and MACs in the modes are configured unless otherwise indicated (see Table 148).

Table 148: Supported Ciphers and MACs

Mode	Ciphers	MACs
Default	*Enabled* by default: aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com *Disabled* by default: 3des-cbc, aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se	*Enabled* by default: hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com; hmac-sha1, hmac-sha2-256, hmac-sha2-512 *Disabled* by default: hmac-md5, hmac-md5-96, hmac-md5-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-sha1-96, hmac-sha1-96-etm@openssh.com
FIPS	aes128-cbc, aes192-cbc, aes256-cbc, aes128-ctr, aes192-ctr, aes256-ctr, rijndael-cbc@lysator.liu.se	hmac-sha1, hmac-sha2-256, hmac-sha2-512
Secure	aes128-ctr, aes192-ctr, aes256-ctr	hmac-sha1; hmac-sha2-256; hmac-sha2-512

By default, weaker ciphers/MACs are disabled, while the remaining ciphers/MACs are enabled (see Default mode in Table 148). This is the default mode of operation. A switch can move from this default mode to FIPS mode. Changing from default mode to FIPS mode requires switch reboot. Until reboot, all the SSH connections use default mode configurations. Since the configuration is already changed by the user (here FIPS mode), updated configuration is displayed. On reboot, only FIPS mode ciphers/MACs are available in SSH for configuration. Again, disabling FIPS mode requires a system reboot. Until reboot, SSH server supports only FIPS mode ciphers/MACs. Since the user configured disabling FIPS mode, the corresponding updated configuration appears. It is recommended to update the FIPS mode once all the operations are completed. None of the security related configurations are supposed to be changed after updating the FIPS mode. This is because the behavior is undefined in those scenarios.

SSH supports Secure-Mode, which is allows only highly secure ciphers/MACs. Secure-Mode is a feature of SSH alone. This has no effect on other modules that use cryptograph like FIPS-mode. Configuring Secure-Mode does not require a switch reboot. The changes are immediate. Secure-mode and FIPS-mode are not mutually exclusive. Secure-Mode supports fewer ciphers/MACs than FIPS mode, so you only have Secure-Mode ciphers/MACs available for configuration with both Secure-Mode and FIPS-Mode are enabled. Internally, Secure-Mode ueses FIPS library for cryptography instead of OpenSSL library. Once Secure-Mode is disabled, the SSH server moves to either default-mode or FIPS-mode, based on FIPS-Mode configuration. However, those ciphers/MACs that are configured before enabling/disabling Secure-Mode are lost. So, for each mode update, SSH server configuration of ciphers/MACs is reset. You have to configure ciphers/MACs each time if you configured SSH server for using particular ciphers/MACs. From the list of available ciphers/MACs, you can configure the required ciphers/MACs, and disable the ciphers/MACs that are not required.

Disabling all the ciphers/MACs does not allow any incoming connection from the clients. This happens because the incoming connections are not able to negotiate for a common cipher/MAC.

Configuring Unapproved Crypto Algorithms

- To configure the required ciphers/MACs with SSHv2, use the following command:
  `configure ssh2 enable [`**`cipher`** `[`*`cipher`* `|`**`all`**`] |`**`mac`** `[ `*`mac`* `|`**`all`**`]]`
- To disable ciphers/Message Authentication Codes (MACs) for use with SSHv2, use the following command:
  `configure ssh2 disable [`**`cipher`** `[`*`cipher`* `|`**`all`**`] |`**`mac`** `[ `*`mac`* `|`**`all`**`]]`
- To display the configured SSHv2 ciphers and MACs, use the following command:
  `show ssh2 {`**`ciphers`** `|` **`macs`**`}`

Enabling/Disabling Ciphers and MACs Configuration Examples

Configuration Example: Enable/Disable Ciphers in Default Mode

```
# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
Key algorithm               :
  Current                   : rsa-sha2-256
  Configured                : rsa-sha2-256
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96
```

```
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
Login grace timeout : 100 seconds

# show ssh2 ciphers
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com

# configure ssh2 disable cipher aes128-cbc

 # show ssh2 ciphers
Ciphers               : aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se, aes128-ctr,
aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com

# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
Key algorithm               :
  Current                   : rsa-sha2-256
  Configured                : rsa-sha2-256
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se, aes128-ctr,
aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
Login grace timeout : 100 seconds

# configure ssh2 enable cipher aes128-cbc
Notice: The configured cipher(s), aes128-cbc, is/are weaker than what is recommended.

# show ssh2 ciphers
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com

# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
Key algorithm               :
  Current                   : rsa-sha2-256
  Configured                : rsa-sha2-256
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
```

```
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
```

Configuration Example: Enable/Disable MACs in Default Mode

```
# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
Login grace timeout : 100 seconds

# show ssh2 macs
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96

# configure ssh2 disable mac hmac-md5-etm@openssh.com

# show ssh2 macs
Macs                  : hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-
sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-
md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-sha1-96, hmac-md5-96

# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
Macs                  : hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-
sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-
md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-sha1-96, hmac-md5-96
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
Login grace timeout : 100 seconds

# configure ssh2 enable mac hmac-md5-etm@openssh.com
```

```
Notice: The configured message authentication code(s), hmac-md5-etm@openssh.com, is/are
weaker than what is recommended.

# show ssh2 macs
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96

# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se,
aes128-ctr, aes192-ctr, aes256-ctr, chacha20-poly1305@openssh.com
Macs                  : hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, hmac-
sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1-96-etm@openssh.com,
hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-
sha1-96, hmac-md5-96
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
Login grace timeout : 100 seconds
```

Configuration Example: Enable/Disable Ciphers in FIPS Mode

```
# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
Login grace timeout   : 120 seconds

# configure ssh2 disable cipher aes128-ctr

# show ssh2 ciphers
Ciphers               : aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc,
rijndael-cbc@lysator.liu.se

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
```

```
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc,
rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

# configure ssh2 enable cipher aes128-ctr

# show ssh2 ciphers
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds
```

Configuration Example: Enable/Disable MACs in FIPS Mode

```
# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

# configure ssh2 disable mac hmac-sha1

# show ssh2 macs
Macs                  : hmac-sha2-256, hmac-sha2-512

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
```

```
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

# configure ssh2 enable mac hmac-sha1

# show ssh2 macs
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds
```

Configuration Example: Enable/Disable Ciphers in Secure Mode

```
# configure ssh2 secure-mode on
Note: All Secure mode Ciphers/MACs will be enabled

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : On
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

# show ssh2 ciphers
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr

# configure ssh2 disable cipher aes128-ctr

# show ssh2 ciphers
Ciphers               : aes192-ctr, aes256-ctr

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
```

```
Secure Mode           : On
Diffie-Hellman Groups : 1 (1024 bits prime), 14 (2048 bits prime)
Idle time             : 60 minutes
Ciphers               : aes192-ctr, aes256-ctr
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

#  configure ssh2 enable cipher aes128-ctr

# show ssh2 ciphers
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : On
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr
Macs                 : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds
```

Configuration Example: Enable/Disable MACs in Secure Mode

```
# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : On
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

# show ssh2 mac
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512

# configure ssh2 disable mac hmac-sha1

# show ssh2 mac
Macs                  : hmac-sha2-256, hmac-sha2-512

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : On
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr
```

```
Macs                  : hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds

# configure ssh2 enable mac hmac-sha1

# show ssh2 mac
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512

# show ssh2
SSH module configuration details:
SSH Access            : Enabled
Key validity          : Valid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : On
Diffie-Hellman Groups :  14 (2048 bits), 16 (4096 bits), 18 (8192 bits))
Idle time             : 60 minutes
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Login grace timeout   : 120 seconds
```

Key Exchange Algorithms—Diffie-Hellman

Diffie-Hellman Overview

Diffie-Hellman (DH) is a public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. This method allows two computer users to generate a shared private key with which they can then exchange information across an insecure channel.

ExtremeXOS supports Diffie-Hellman group 1, 14, 16, and 18 as part of the key exchange algorithms. In compliance with the Network Device collaborative Protection Profile (NDPP) of Common Criteria, the TSF (Target Security Function) ensures that diffie-hellman-group14-sha1 is the only minimal allowed key exchange method used for the SSH protocol.

ExtremeXOS uses group 14 by default as the minimal supported Diffie-Hellman group as 14 to avoid using the weaker Diffie-Hellman group 1 in an SSH server.

Configuring the Diffie-Hellman Minimal Supported Group

- To configure the minimal supported Diffie-Hellman group, use the following command:

  `configure ssh2` **`dh-group minimum`** [**`1`** | **`14`** | **`16`** | **`18`**]

- To show the SSH configuration details, including minimal supported Diffie-Hellman group, use the following command:

  `show ssh2`

Diffie-Hellman Minimal Supported Group Configuration Examples

Configuration Example: DH-16 Key-Exchange Algorithm

```
# show ssh2
SSH module configuration details:
```

```
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss

# configure ssh2 dh-group minimum 16

# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
```

Configuration Example: DH-1 Key-Exchange Algorithm

```
# configure ssh2 dh-group minimum 1
Notice: The configured key exchange algorithm(s), DH group 1, is/are weaker than what is
recommended.

# show ssh2
SSH module configuration details:
SSH Access            : Disabled
Key validity          : Invalid
Key type              : RSA 2048
TCP port              : 22
VR                    : all
Access profile        : not set
Secure Mode           : Off
Diffie-Hellman Groups : 1 (1024 bits), 14 (2048 bits), 16 (4096 bits), 18 (8192 bits)
Max Auth Tries        : 3
Idle time             : 60 minutes
Rekey Interval        : 4096 MB and no time limit
Ciphers               : aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc,
aes256-cbc, rijndael-cbc@lysator.liu.se
Macs                  : hmac-sha1, hmac-sha2-256, hmac-sha2-512
Public key algorithms : ssh-rsa, ssh-dss, x509v3-sign-rsa, x509v3-sign-dss
```

Secure Socket Layer

Secure Socket Layer (SSLv3) is a feature of ExtremeXOS that allows you to authenticate and encrypt data over an SSL connection to provide secure communication.

The existing web server in ExtremeXOS allows HTTP clients to access the network login page. By using HTTPS on the web server, clients securely access the network login page using an HTTPS enabled web browser. Since SSL encrypts the data exchanged between the server and the client, you protect your data, including network login credentials, from unwanted exposure.

HTTPS access is provided through SSL and the Transport Layer Security (TLS1.2). These protocols enable clients to verify server authenticity to prevent network intruders.

You must upload or generate a certificate for SSL server use. Before you can upload a certificate, you must purchase and obtain an SSL certificate from an Internet security vendor. The following security algorithms are supported:

- RSA for public key cryptography (generation of certificate and public-private key pair, certificate signing). RSA key size between 1,024 and 4,096 bits.
- Symmetric ciphers (for data encryption): RC4.
- Message Authentication Code (MAC) algorithms: RSA Data Security, Inc. *MD5* Message-Digest Algorithm and SHA.

Version 32.3 adds a security enhancement that auto-generates an SSL certificate on enabling web HTTPS if an SSL certificate is not already created via the CLI. The certificate is generated with default configurations for the following:

- Key length
- Country-code
- Organization name
- Common name
- State
- Locality
- Organization unit
- Email

Once the certificate is successfully generated, the HTTPS login will be granted. The generation and validation of the certificate and key behaves in the same way as those that are generated through the command line.

Enabling and Disabling SSL

This section describes how to enable and disable SSL on your switch. By default, HTTPs is enabled on the switch.

To use SSL with web-based login (secure HTTP access, HTTPS) you must specify the HTTPS protocol when configuring the redirect URL.

- To enable SSL and allow secure HTTP (HTTPS) access on the default port (443), use the following command:

  ```
  enable web https
  ```

- To disable SSL and HTTPS, use the following command:

  ```
  disable web https
  ```

Creating Self-Signed Certificates and Private Keys

When you generate a certificate, the certificate is stored in the configuration file, and the private key is stored in the EEPROM. The certificate generated is in PEM format. By default ExtremeXOS uses the SHA-512 hashing algorithm to create the certificate. The certificate hashing algorithm can be configured using command `configure ssl certificate hash-algorithm` *`hash-algorithm`*. ExtremeXOS supports *MD5*, SHA-224, SHA-256, SHA-384 and SHA-512. The configured algorithm are used to create certificates from next time onwards. Use the `show ssl` command to check the currently configured Signature hashing algorithm.

- To create a self-signed certificate and private key that can be saved in the EEPROM, use the following command:

  `configure ssl certificate privkeylen` *`length`* **`country`** *`code`* **`organization`** *`org_name`* **`common-name`** *`name`*

  Make sure to specify the following:

  - Country code (maximum size of 2 characters)
  - Organization name (maximum size of 64 characters)
  - Common name (maximum size of 64)

  Any existing certificate and private key is overwritten.

  The size of the certificate depends on the RSA key length (**`privkeylen`**) and the length of the other parameters (country, organization name, and so forth) supplied by the user. For an RSA key length of 4,096, the certificate length is approximately 2 Kb, and the private key length is approximately 3 Kb.

Downloading a Certificate Key from a TFTP Server

You can download a certificate key from files stored in a TFTP server. If the operation is successful, any existing certificate is overwritten. After a successful download, the software attempts to match the public key in the certificate against the private key stored. If the private and public keys do not match, the switch displays a warning message similar to the following: `Warning: The Private Key does not match with the Public Key in the certificate.` This warning acts as a reminder to also download the private key.

Downloaded certificates and keys are not saved across switch reboots unless you save your current switch configuration. After you use the `save` command, the downloaded

certificate is stored in the configuration file and the private key is stored in the EEPROM.

- To download a certificate key from files stored in a TFTP server, use the following command:

  ```
  download ssl ipaddress certificate {ssl-cert | trusted-ca | ocsp-signature-ca | {csr-cert {ocsp [on | off]}} file_name
  ```

  Note
  For security measures, you can only download a certificate key in the *VR-Mgmt*.

- To see whether the private key matches with the public key stored in the certificate, use the following command:

  ```
  show ssl {detail}
  ```

  This command also displays:

  - HTTPS port configured. This is the port on which the clients will connect.
  - Length of the RSA key (the number of bits used to generate the private key).
  - Basic information about the stored certificate.

Downloading a Private Key from a TFTP Server

For security reasons, when downloading private keys, we recommend obtaining a pre-generated key rather than downloading a private key from a TFTP server. See Configuring Pregenerated Certificates and Keys on page 1678 for more information.

- To download a private key from files stored in a TFTP server, use the following command:

  ```
  download ssl ipaddress privkey key_file
  ```

  If the operation is successful, the existing private key is overwritten. After the download is successful, a check is performed to find out whether the private key downloaded matches the public key stored in the certificate. If the private and public keys do not match, the switch displays a warning message similar to the following: `Warning: The Private Key does not match with the Public Key in the certificate.` This warning acts as a reminder to also download the corresponding certificate.

Downloaded certificates and keys are not saved across switch reboots unless you save your current switch configuration. After you use the `save` command, the downloaded certificate is stored in the configuration file and the private key is stored in the EEPROM.

Configuring Pregenerated Certificates and Keys

- To get the pregenerated certificate from the user, use the following command:

  ```
  configure ssl certificate pregenerated
  ```

  You can copy and paste the certificate into the command line followed by a blank line to end the command.

This command is also used when downloading or uploading the configuration. Do not modify the certificate stored in the uploaded configuration file because the certificate is signed using the issuer's private key.

The certificate and private key file should be in PEM format and generated using RSA as the cryptography algorithm.

- To get the pregenerated private key from the user, use the following command:

  ```
  configure ssl privkey pregenerated
  ```

  You can copy and paste the key into the command line followed by a blank line to end the command.

  This command is also used when downloading or uploading the configuration. The private key is stored in the EEPROM.

  The certificate and private key file should be in PEM format and generated using RSA as the cryptography algorithm.

Support for x509v3-rsa2048-sha256 host key

Version 33.2.1 adds support for the x509v3-rsa2048-sha256 host key. The SSH server can present its x509v3 certificate (host key) to clients, and external SSH clients can authenticate the SSH server with the server's x509v3 certificate.

Supported Platforms

All platforms.

New CLI Command

```
configure ssh2 x509v3 key algorithm
```

Example

The following command configures the x509v3-rsa2048-sha256 key algorithm:

```
configure ssh2 enable x509v3 key algorithm x509v3-rsa2048-sha256
```

Modified CLI Command

The configure ssl csr, download ssl certificate, show ssl, and show ssl csr commands add the ssh keyword to the syntax:

- configure ssl csr {ssh} privkeylen *length* country *code* organization *org_name* common-name *name*
- download ssl *ip_address* certificate csr-cert {ssh} *file_name* {ocsp [on | off]
- show ssl {manufacturing} {certificate {ssh} | detail}

Creating Certificate Signing Requests and Private Keys

Secure Socket Layer (SSL) allows you to:

- Generate self-signed certificates, which generate private keys and self-signed X509 certificates.

- Download SSL private key/certificate using `download ssl` *ipaddress* **certificate** {**ssl-cert** | **trusted-ca** | **ocsp-signature-ca** | {**csr-cert** {**ocsp** [**on** | **off**]}} *file_name* command (generally used to download CA signed certificate).
- Obtain an SSL private-key/certificate using the `configure` **ssl certificate pregenerated**{ {**csr-cert**}**pregenerated** {**ocsp** {**on** | **off**}}} command (generally used to obtain the CA signed certificate for copying).

Version 33.4.1 adds support for downloading TLS certificates using both VR-Default and user-defined virtual routers, in addition to the previously supported VR-Mgmt. This enhancement improves deployment flexibility in environments where out-of-band (OoB) management is not used.

Additionally, you can create certificate signing requests (CSRs)/private key pairs. The CSR can then be taken to a Certificate Authority (CA) for signing. The CA then provides the signed certificate, which can be downloaded to the switch using either of the commands listed previously.

To create a CSR, use the following command:

`configure` **ssl csr privkeylen** *length* **country** *code* **organization** *org_name* **common-name** *name*

To view the CSR any time after creating it, use the following command:

`show ssl csr`

Note
For enhanced security, the minimum private key length is 2,048 (previously it was 1,024). This length is enforced in both private key/self-signed certificate pairs and private key/CSR pairs.

Displaying SSL Information

- To display whether the switch has a valid private and public key pair and the state of HTTPS access, use the following command:
  `show ssl {[`**trusted-ca** | **ocsp-signature-ca**`] [`*file_name* | **all**`]} {`**manufacturing**`}{`**certificate** | **detail**`}`
- To display a certificate signing request (CSR), use the following command:
  `show ssl csr`

Using Public-Key Infrastructure (PKI) in Your Network

The ExtremeXOS implementation of public-key infrastructure (PKI) supports the secure authentication of Syslog server and SSH client to an Extreme Networks XOS device using an X.509 certificate. Below are primary aspects of a PKI configuration (for

more information about each command listed in this topic, see Switch Engine v33.6.1 Command References):

> **Note**
> All the certificates mentioned below should be in PEM format.

- Trusted CA—The X509v3 certificates of Certificate Authority (CA) should be downloaded using the CLI: `download ssl` *`ipaddress`* **`certificate`** {**`ssl-cert`** | **`trusted-ca`** | **`ocsp-signature-ca`** | {**`csr-cert`** {**`ocsp`** [**`on`** | **`off`**]}} *`file_name`* using the **`trusted-ca`** option. The CA certificate must satisfy the following criteria to successfully download:
  - Basic constraints: CA = true
  - Key usage must contain: KeyCertSign
- Peer Certificate—X509v3 certificate of the peer, signed by one of the above trusted CAs. The following criteria must be met for successful authentication:
  - Syslog server certificate: Extended key usage must contain 'Server Authentication'
  - SSH Client certificate:
    - Common name (CN) of the certificate subject must be same as the username with which SSH session is tried out.
    - Extended key usage must contain 'Client Authentication.'
- OCSP—Online Certificate Status Protocol used to find the revocation status of the peer certificate in the following scenarios:
  - Syslog server certificate's OCSP status is identified when a TLS session is to be established with the Syslog server. Only if the OCSP status is GOOD is the session established.
  - SSH Client certificate's OCSP status is identified as part of authentication. Only if the OCSP status is GOOD is the session established.

  > **Note**
  > OCSP processes intermediate CA certificates iteratively, one by one.

  The OCSP Server's address must be configured in the Authority Information Access (AIA) of the peer certificate. Otherwise, the PKI authentication fails. The supported OCSP responder models are: common issuer model, delegated trusted responder model, trusted responder model.
- OCSP Signature CA—The certificates of OCSP responders must be configured as OCSP signature CA and trusted CA. To support Trusted Responder Model (TRM) of OCSP, the X509v3 certificate of the OCSP responder should be downloaded using the CLI: `download ssl` *`ipaddress`* **`certificate`** {**`ssl-cert`** | **`trusted-ca`** | **`ocsp-signature-ca`** | {**`csr-cert`** {**`ocsp`** [**`on`** | **`off`**]}} *`file_name`* using the ocsp-signature-ca and trusted-ca options. For example:

  ```
  download ssl 10.120.89.79 certificate trusted-ca cacert.pem
  ```

  The OCSP signature CA is only required for TRM; it is not used for DTM and common issuer. This certificate must contain a trusted use extension that permits OCSP signing. A "trusted use extension" can be appended to a certificate using OpenSSL.

The following example appends a trusted use extension specifying an original file and the trusted file: `ocsp-sig-ca.pem` is the original certificate file and the output file `trusted-ocsp-sig-ca.pem` is the trusted file: % openssl x509 -in ocsp-sig-ca.pem -addtrust OCSPSigning -out trusted-ocspsig- ca.pem

The following is an example of an original certificate followed by the OpenSSL command output trusted certificate:

```
-----BEGIN CERTIFICATE-----
MIICgTCCAeqgAwIBAgIJAMng4JQ0MOeIMA0GCSqGSIb3DQEBBQUAMGAxCzAJBgNV
 BAYTAlVTMRIwEAYDVQQKEwlFbnRlcmFzeXMxDDAKBgNVBAsTA0RvRDEMMAoGA1UE
 CxMDUEtJMSEwHwYDVQQDExhFc3lzIEpJVEMgT0NTUCBSZXNwb25kZXIwHhcNMTIw
 MjE3MTg0MzEwWhcNMjIwMjE0MTg0MzEwWjBgMQswCQYDVQQGEwJVUzESMBAGA1UE
 ChMJRW50ZXJhc3lzMQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEhMB8GA1UE
 AxMYRXN5cyBKSVRDIE9DU1AgUmVzcG9uZGVyMIGfMA0GCSqGSIb3DQEBAQUAA4GN
 ADCBiQKBgQCuyC9QHBpP/n6aOS+Cx0mbgsQTS1LAUUCwxjvJdILGVfdjFB8PKG+o
 W4jm7FKuRHR7uzBvAFzD9DbVkziHl2yIsy4SeiSBTQpNvHPjvUcec3rTlw7saiTw
 B+CTqEm1pxcEdRKTvawK2k1ujHML1MABP2CA3SEptO+Ude4UkXMBywIDAQABo0Mw
 QTAdBgNVHQ4EFgQUYFhsLiklZh0riJ1Hg7d4HPcLlBUwCwYDVR0PBAQDAgGGMBMG
 A1UdJQQMMAoGCCsGAQUFBwMJMA0GCSqGSIb3DQEBBQUAA4GBADU4aQ6f8pHWLd7z
 vZ8pJ8e8UCvKok1LmdXbax5TBonyyLmb7AjLrOWjZ7LKSufJL1KOBsetd5Q49LFK
 h70V2fRWpGNQszpAV60WfidkNvQ0koZczEjYRQOCtMDUqxMHxsMv2MLEVE9QuGLt
 +NWjeeF03E1DT3C4mnbVsTyWPZij
 -----END CERTIFICATE-----
-----BEGIN TRUSTED CERTIFICATE-----
MIICgTCCAeqgAwIBAgIJAMng4JQ0MOeIMA0GCSqGSIb3DQEBBQUAMGAxCzAJBgNV
 BAYTAlVTMRIwEAYDVQQKEwlFbnRlcmFzeXMxDDAKBgNVBAsTA0RvRDEMMAoGA1UE
 CxMDUEtJMSEwHwYDVQQDExhFc3lzIEpJVEMgT0NTUCBSZXNwb25kZXIwHhcNMTIw
 MjE3MTg0MzEwWhcNMjIwMjE0MTg0MzEwWjBgMQswCQYDVQQGEwJVUzESMBAGA1UE
 ChMJRW50ZXJhc3lzMQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEhMB8GA1UE
 AxMYRXN5cyBKSVRDIE9DU1AgUmVzcG9uZGVyMIGfMA0GCSqGSIb3DQEBAQUAA4GN
 ADCBiQKBgQCuyC9QHBpP/n6aOS+Cx0mbgsQTS1LAUUCwxjvJdILGVfdjFB8PKG+o
 W4jm7FKuRHR7uzBvAFzD9DbVkziHl2yIsy4SeiSBTQpNvHPjvUcec3rTlw7saiTw
 B+CTqEm1pxcEdRKTvawK2k1ujHML1MABP2CA3SEptO+Ude4UkXMBywIDAQABo0Mw
 QTAdBgNVHQ4EFgQUYFhsLiklZh0riJ1Hg7d4HPcLlBUwCwYDVR0PBAQDAgGGMBMG
 A1UdJQQMMAoGCCsGAQUFBwMJMA0GCSqGSIb3DQEBBQUAA4GBADU4aQ6f8pHWLd7z
 vZ8pJ8e8UCvKok1LmdXbax5TBonyyLmb7AjLrOWjZ7LKSufJL1KOBsetd5Q49LFK
 h70V2fRWpGNQszpAV60WfidkNvQ0koZczEjYRQOCtMDUqxMHxsMv2MLEVE9QuGLt
 +NWjeeF03E1DT3C4mnbVsTyWPZijMAwwCgYIKwYBBQUHAwk=
 -----END TRUSTED CERTIFICATE-----
```

- ExtremeXOS X509v3 certificate—The certificate of the ExtremeXOS device. This is shared to the Syslog server to perform authentication there. Use the commands: `download ssl` *`ipaddress`* **`certificate`** {**`ssl-cert`** | **`trusted-ca`** | **`ocsp-signature-ca`** | {**`csr-cert`** {**`ocsp`** [**`on`** | **`off`**]}} *`file_name`* with the **`ssl-cert`** option, and `download ssl` *`ipaddress`* **`privkey`** *`key_file`* .
- Beginning with software release 32.2, you can configure SSH x509v3 OCSP attributes (enable/disable, nonce, override, and ocsp-nocheck, respectively) using the following commands:
  - `configure ssh2` **`x509v3 ocsp`** [**`on`** | **`off`**]
  - `configure ssh2` **`x509v3 ocsp nonce`** [**`on`** | **`off`**]
  - `configure ssh2` **`x509v3 ocsp override`** [*`url`* | **`none`**]
  - `configure ssh2` **`x509v3 ocsp signer ocsp-nocheck`** [**`on`** | **`off`**]

OCSP nonce cryptographically binds an OCSP request and an OCSP response with an `id-pkix-ocsp-nonce` extension to prevent replay attacks.

OCSP override configures one HTTP Online Certificate Status Protocol (OCSP) override URL for an SSH2 x509v3 server.

When OCSP-nocheck is done for a peer certificate, ExtremeXOS sends the OCSP request to the OCSP server. The OCSP response is signed by the OCSP responder/signer. The response also comes along with the certificate of the OCSP signer. When ExtremeXOS receives the response, it only verifies that the status of the peer certificate is not revoked.

Setting Up PKI

The following is the sequential workflow involved in the session establishment using PKI:

1. Generate the involved X509v3 certificates: CA certificates, OCSP Signature CA certificate, Peer certificate (for example: Syslog server or SSH client), ExtremeXOS device certificate.
2. Download the CA certificates and OCSP Signature CA certificates to the ExtremeXOS device.
3. Download the ExtremeXOS device certificate and key to ExtremeXOS device (required for establishing TLS session with Syslog server).
4. Configure the peer (Syslog server or SSH client) as required to use its own X509v3 certificate in the connection request.
5. Initiate the connection request from peer (Syslog server or SSH client) to ExtremeXOS device.
6. The ExtremeXOS device performs the following tasks on the received peer's certificate and accepts or rejects the connection request:
   - Certificate chain verification
   - Validity checks on certificate extensions
   - OCSP

PKI Limitations

- All certificates should be in PEM format files.
- Downloading CA certificate chain is not supported.
- Individual CA certificates in a certificate chain should be downloaded one-by-one using the following command: `download ssl` *`ipaddress`* **`certificate`** {**`ssl-cert`** | **`trusted-ca`** | **`ocsp-signature-ca`** | {**`csr-cert`** {**`ocsp`** [**`on`** | **`off`**]}} *`file_name`*
- Downloading CA certificate of size greater than 7.5KB is not recommended.
- Certification Revocation Lists (CRLs)—not supported.
- OCSP stapling—not supported.
- Nonce is always disabled in OCSP request.
- OCSP is not done for the OCSP responder certificate. Therefore, the OCSP responder certificate should satisfy any of following criteria, failing which the OCSP response is rejected:
  - OCSP responder certificate should be self-signed, OR
  - OCSP responder certificate should contain `id-pkix-ocsp-nocheck` extension.

ExtremeXOS Image Integrity Check

If the ExtremeXOS image integrity check feature is enabled, during bootup, the system checks the integrity of the ExtremeXOS image, and notifies you if it has been compromised or not (error is reported in the Syslog).

Configuring ExtremeXOS Image Integrity Check

To enable or disable the ExtremeXOS image integrity check feature, use the following command:

```
configure switch integrity-check image [on | off]
```

To view the status and configuration of the ExtremeXOS image integrity check feature, use the following command:

```
show switch management
```

This command shows one of the following values:

- On (Valid)—the feature is enabled, and the check was successful.
- On (Not Checked)—the feature is enabled, but the image has not been checked. This occurs immediately after enabling the feature, but before rebooting, which will initiate the integrity check.
- On (Invalid)—the feature is enabled, and the check failed. The ExtremeXOS image is corrupted.
- On (Failed)—the feature is enabled, and the integrity check failed to run due to failures.
- Off—the feature is disabled.

ExtremeXOS Image Integrity Check Syslog Messages

If an integrity failure occurs, the following error message is logged:

```
<ERROR> INTEGRITY: File <file-path> has invalid hash; expected <expected-hash> actual
<calculate-hash>.
```

If all critical files have expected hash values (passed integrity check), the following messages are logged:

```
<INFO> INTEGRITY: INTEGRITY-CHECK-VALID
<INFO> INTEGRITY: Image Integrity verification passed (<hash-algorithm>).
```

If one or more critical files have unexpected hash values (failed integrity check), the following error messages are logged:

```
<INFO> INTEGRITY: INTEGRITY-CHECK-INVALID
<INFO> INTEGRITY: Image Integrity verification failed (<hash-algorithm>).
```

Secure Boot

Secure Boot is a mechanism to ensure the integrity of firmware and software running on a hardware platform by establishing a chain-of-trust relationship in the boot process. The chain-of-trust is established by cryptographic checks at each stage of the

boot process to validate the integrity and authenticity of the next stage before it can execute.

The first link in the chain-of-trust is called the “Hardware Root of Trust” (HWROT), which is always trusted and protected against any alterations once programmed. For this version of Secure Boot, the chain-of-trust is established between HWROT, bootloader(s) (ARM systems)/BIOS (X86 systems). The HWROT comprises hardware components ASP NOR Flash, TPM, the firmware ‘Secondary Program Loader’ (SPL), and the recovery bootloader.

If the UBOOT image verification fails during boot-up, the switch halts and enters the recovery bootloader.

Supported Platforms

ExtremeSwitching 5420, 5520, 5720, 7520, and 7720 series switches.

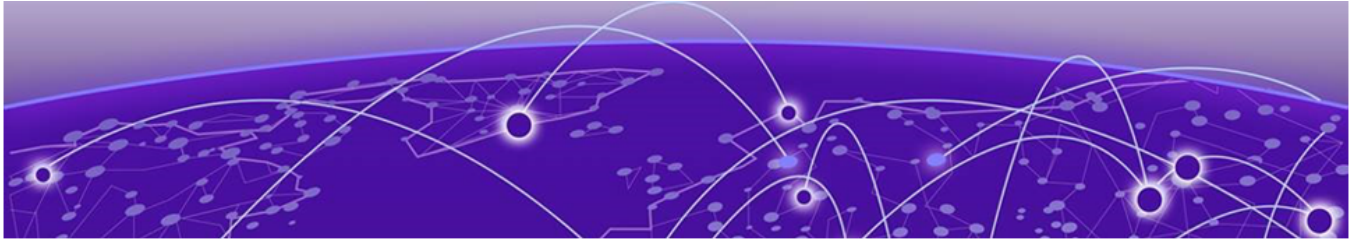
Verifying Secure Boot

The following two commands show you the Secure Boot status in the **Trusted Delivery** field:

```
show switch {detail}

show system
```

The **Trusted Delivery** field has the following states:

- Boot Image Verification Failed
- Boot Image Verified



Service Probe

Service Probe Overview on page 1686
Configuring Service Probes on page 1691
Running Service Probe Actions on page 1692
Displaying Service Probe Information on page 1694
Configuration Examples on page 1695

This chapter discusses the Service Probe feature, which emulates a client device connected to your network to help you test and debug network functionality from an end-user perspective. Using a Service Probe, you can verify DHCP configuration, test Layer 3 connectivity via ping, validate DNS server resolution, and confirm default gateway reachability — all from within the device itself.

This chapter provides an overview of the feature's capabilities, implementation behavior, and example CLI commands.

Service Probe Overview

When you troubleshoot connectivity or service issues from the switch, you're seeing the network from the switch's perspective — not the user's. Latency measurements, DHCP behavior, and DNS resolution can all look different depending on where in the network you're standing. A **Service Probe** bridges that gap.

A Service Probe emulates a client device connected to a VLAN on your switch. It operates completely outside ExtremeXOS's normal networking domain, so it sees the network the same way a device plugged into a front panel port would. This makes it an effective tool for validating end-user experience, catching problems that don't show up from the switch itself, and debugging intermittent issues without needing a physical test device.

Once created, a Service Probe can:

- Obtain an IP address via DHCP, verifying DHCP server reachability and response.
- Test L3 connectivity to any destination using ping.
- Validate default gateway reachability using ARP.
- Resolve hostnames using the probe's own DNS configuration.
- Execute shell commands or Python scripts within the probe's isolated network context.

Service Probe was introduced in version 33.5.1. Shell and Python code execution, enhanced DNS querying, and asynchronous command handling were added in version 33.6.1.

How Service Probes Work

Each Service Probe is implemented as a Linux network namespace containing a single `macvlan`-type network device, called the *service interface*. This service interface is linked to the network device that represents the specified VLAN, but lives in a completely separate namespace from the OS itself.

From the network's perspective, a Service Probe looks and behaves like any other device connected to that VLAN — it has its own MAC address, responds to ARP, can send and receive traffic, and is visible to neighboring devices. Multiple probes can share the same VLAN; each still gets its own namespace and service interface.

The probe's MAC address is registered with the OS slow path using the VMAC infrastructure, and programmed into hardware on all stack nodes via the FDB filter mechanism so traffic is forwarded correctly to the CPU.

MAC Address Allocation

Each Service Probe is assigned a unique MAC address that is different from the switch MAC.

- On **universal hardware**, MAC addresses are drawn from the device's dedicated 1,024-address pool. Up to 16 are allocated per VLAN for Service Probes.
- On **non-universal hardware, VMs, and Digital Twin**, the two available LAA addresses are used (`0A:xx:xx:xx:xx:xx` and `0E:xx:xx:xx:xx:xx`), which limits probes to two per VLAN.

MAC addresses are re-used across VLANs — the same MAC can appear on probes attached to different VLANs at the same time.

IP Address Configuration

A Service Probe gets its IP configuration in one of two ways:

Dynamic (DHCP): The probe runs a DHCP client (`udhcpc`) inside its namespace, requesting an address, subnet mask, gateway, and DNS servers from the network. This is a real DHCP exchange from the perspective of the DHCP server — it's an effective way to verify that DHCP is working for clients on that VLAN.

Static: An IP address, subnet mask, and gateway are assigned when the probe is created and remain fixed.

When a dynamic probe is created, it immediately attempts a synchronous DHCP request. To keep this initial delay short, the DHCP client uses the following timing:

Parameter	Value	Description
Retries	2	Number of DHCP requests sent per attempt cycle
Timeout	2 seconds	Time to wait for a response before retrying
Try-again interval	30 seconds	Wait time before starting a new attempt cycle if no response was received

If no offer is received during the initial attempt, the probe is still created and the DHCP client continues retrying in the background. You can also manually trigger a renew or release at any time.

Note
Service Probes cannot be associated with the Management VLAN.

DNS Name Server Support

Each probe namespace maintains its own `/etc/resolv.conf`, giving every probe independent DNS configuration. Up to three name servers can be configured — the maximum `libc` will use. Name servers can come from:

- **DHCP:** Automatically applied when using dynamic IP configuration.
- **Manual configuration:** Required for static IP probes, and can also supplement or override DHCP-provided servers.

In version 33.6.1 and later, you can direct a DNS query to a specific server (primary, secondary, or tertiary), all configured servers at once, or the default system resolver. The results of the most recent query to each server are stored and shown in `show service-probe detail`.

Gateway Validation

The `run service-probe query gateway` command tests whether the probe's configured gateway is reachable by sending ARP requests from inside the probe's namespace using `arping`.

Parameter	Value
Retries	3 ARP requests
Timeout per request	2 seconds

This command runs synchronously and completes in at most roughly 6 seconds. The most recent result is stored and shown in `show service-probe detail`.

Ping

The `run service-probe ping` command sends ICMP echo requests to a hostname or IP address from within the probe's network context. The result is reported as **Pass**, **Fail**, or **Not Completed**.

Parameter	Value
Count	3 packets
Wait per packet	2 seconds

Note
A complete failure (all three pings timing out) takes up to 6 seconds. In version 33.6.1 and later, ping runs asynchronously so it does not block the CLI while waiting.

Code Execution

Introduced in version 33.6.1. The `run service-probe shell` and `run service-probe python` commands let you run arbitrary shell commands or Python scripts from inside the probe's network namespace. This is useful when you need to go beyond ping — for example, running `traceroute`, testing HTTP connectivity, or executing a custom diagnostic script that relies on the probe's IP stack and DNS configuration.

Results include:

Field	Description
`normalExit`	Whether the process exited cleanly
`exitStatus`	The process exit code
`output`	Combined stdout/stderr (up to 7,000 bytes)
`outputTruncated`	True if output exceeded the 7,000-byte buffer

To make scripts easier to write, the OS automatically sets environment variables describing the probe's current state at the time of execution:

```
SP_VLAN_NAME='uplink' SP_VLAN_ID='3500' SP_VLAN_UP='0' SP_IP_ADDR='172.16.1.99'
SP_IP_ADDRMASK='172.16.1.99/24' SP_IP_NETMASK='255.255.255.0' SP_IP_GATEWAY='172.16.1.99'
SP_DNS='11.100.100.1,172.16.1.98' SP_DYNAMIC='0' SP_MAC='0a:11:88:fe:ec:36'
```

Asynchronous Command Execution

In version 33.6.1 and later, ping, query DNS, shell, and Python commands run asynchronously. The workflow is:

1. You issue a command. A request ID is returned immediately.
2. The OS polls for the result in the background and displays it when ready.

3. Once retrieved, the request is automatically removed.
4. If a result is never retrieved, it is automatically removed 10 minutes after the command's maximum expected run time.

Up to 10 actions can run concurrently by default.

Command timeouts:

Command	Check Interval	Max Run Time
Query DNS	1 second	8 seconds
Ping	1 second	10 seconds
Shell	1 second	15 seconds (configurable per request)
Python	1 second	15 seconds (configurable per request)

Resource Limits and Security

All Service Probe commands run inside a dedicated `SvcProbe` cgroup, isolating their resource consumption from the rest of the system:

Resource	Limit
CPU	5% maximum (drawn from the Other cgroup allocation)
Memory	5% maximum

Commands run as the `admin` user, not `root`. Only the OS accounts with admin privileges can issue `run` commands or make POST requests to the NOS-API for Service Probes, so the shell and Python execution features do not introduce any new privilege escalation paths beyond what already exists.

Redundancy and Persistence

Service Probes are not saved to the switch configuration and do not survive a reboot or a netTools process restart. After either event, probes must be recreated.

In stacking environments, a subset of probe data is checkpointed to the backup node — not for probe functionality, but so that FDB filters can be cleaned up correctly if the backup becomes primary. When a probe is created, the backup node receives just enough data to create the corresponding FDB filter. When a probe is deleted, the backup node removes it. On failover, the new primary uses this checkpointed data to remove any hardware programming left over from the previous primary.

Supported Platforms

Service Probe is supported on all Switch Engine platforms. The maximum number of probes per VLAN depends on the platform:

Platform Type	Max Probes per VLAN	Reason
Universal hardware	16	16 MAC addresses allocated from the device's 1,024-address pool
VM, Digital Twin	2	Limited to 2 available Locally Administered Addresses (LAA)

There is no hard limit on the total number of Service Probes across all VLANs. A tested maximum is published in the limits documentation for each release.

Limitations

Before configuring Service Probes, be aware of the following restrictions:

- **On-switch DHCP server** — Not supported. The DHCP server requires a port association, which Service Probes do not have.
- **DHCP relay** — Not supported.
- **Pinging the probe from the OS** — You cannot ping a Service Probe's IP address from the management plane via the linked VLAN interface.
- **Management VLAN** — Service Probes cannot be linked to the Mgmt VLAN.
- **VLAN state changes** — If the linked VLAN goes down or comes back up, the probe is unaffected. DHCP configuration is not cleared or renewed automatically. The current VLAN state is shown in `show service-probe` for reference.
- **IPv6** — Not supported in this release.

Configuring Service Probes

The following sections describe how to configure Service Probes.

Creating a Service Probe

Specify an instance ID between 1 and 9999, and associate the probe with a VLAN by name or ID. If you do not specify IP parameters, the probe uses DHCP.

Syntax:

```
create service-probe <instance_id> [{vlan} <vlan_name> | vlan <vlan_id>]
  {ipaddress [<ipaddress> <netmask> | <ipNetmask>] <gateway>}
```

Create a dynamic (DHCP) probe:

```
create service-probe 1 vlan corp
```

Create a static IP probe:

```
create service-probe 5 vlan corp ipaddress 192.168.100.90/24 192.168.100.1
```

Configuring DNS Name Servers

For static IP probes, add name servers manually. You can also use this command to add or replace servers on dynamic probes. A maximum of three servers can be configured per probe.

Add name servers:

```
configure service-probe <instance_id> add name-server <ip1> {<ip2> {<ip3>}}
```

Remove specific name servers:

```
configure service-probe <instance_id> delete name-server <ip1> {<ip2> {<ip3>}}
```

Remove all name servers:

```
configure service-probe <instance_id> delete name-server all-name-servers
```

Use `all` in place of `<instance_id>` to apply the change to every Service Probe.

Renewing or Releasing a DHCP Address

Renew:

```
run service-probe <instance_id> renew
```

Release:

```
run service-probe <instance_id> release
```

Use `all` to act on all dynamic probes at once. These commands return an error if the target probe has a static IP address.

Deleting a Service Probe

```
delete service-probe <instance_id>
delete service-probe all
```

Running Service Probe Actions

The following sections describe running Service Probe actions.

Querying the Gateway

Tests whether the probe's configured gateway is reachable using ARP. This command runs synchronously.

```
run service-probe <instance_id> query gateway
```

Example output:

```
Pass: Gateway 192.168.100.10 for Service Probe with ID 1 is reachable
Fail: Gateway 192.168.200.11 for Service Probe with ID 100 is not reachable
```

Pinging a Host

Pings an IP address or hostname from within the probe's network context.

```
run service-probe <instance_id> ping <host>
```

Example output:

```
Pass: Ping to 11.100.100.1 using Service Probe with ID 1 succeeded
Fail: Ping to 11.100.100.10 using Service Probe with ID 1 failed
Not Completed: Ping to 11.100.100.99 using Service Probe with ID 99 was not completed
```

Querying DNS

Resolves a hostname using the probe's DNS configuration.

Version 33.5.x:

```
run service-probe <instance_id> query dns <fqdn>
```

Version 33.6.1 and later — optionally target a specific server:

```
run service-probe <instance_id> query dns {primary | secondary | tertiary | default}
{<fqdn>}
```

- Omit the server keyword to query all configured servers in sequence.
- Use `default` to use the standard system resolver (`glibc`).
- Omit the FQDN to use a default test hostname.

Example output:

```
DNS query using Service Probe with ID 1:
Pass: DNS query for www.extremenetworks.com using primary server 11.100.100.1 succeeded
Fail: DNS query for www.extremenetworks.com using secondary server 8.8.8.8 failed
```

Running Shell Commands

Available in version 33.6.1 and later. Executes a shell command string inside the probe's namespace. When you run the command, you are prompted to enter the command string interactively.

```
run service-probe <instance_id> shell
```

Example output:

```
Command:
traceroute 11.100.100.1
Exit Status: 0
...output follows:
traceroute to 11.100.100.1 (11.100.100.1), 30 hops max, 46 byte packets
 1  192.168.1.1 (192.168.1.1)  0.547 ms  0.343 ms  0.321 ms
 2  110.110.110.2 (110.110.110.2)  0.364 ms  88.943 ms  0.375 ms
 3  11.100.100.1 (11.100.100.1)  0.287 ms  0.271 ms  0.247 ms
```

Running Python Scripts

Available in version 33.6.1 and later. Executes a Python script file inside the probe's namespace.

```
run service-probe <instance_id> python <script_file>
```

Example output:

```
Exit Status: 0
...output follows:
Username: admin, User ID: 1001
Group Name: admin, Group ID: 1001
```

Displaying Service Probe Information

Display a summary of all Service Probes:

```
show service-probe
```

Example output:

```
 ID  VLAN        In-Use MAC         In-Use IP            In-Use Gateway  Flags
---- ----------- ------------------ -------------------- --------------- -----
  1  corp        40:88:2f:df:e4:01  192.168.100.102/24   192.168.100.10  D-NRV
  2  corp        40:88:2f:df:e4:02  192.168.100.103/24   192.168.100.10  D-NRV
  5  corp        40:88:2f:df:e4:03  192.168.100.90/24    192.168.100.10  -nNRV
 18  corp100     40:88:2f:df:e4:04  192.168.200.93/24    192.168.200.10  -nNRV

Flags: (D) Dynamic IP, (n) DNS Name Servers Configured, (N) DNS Name Servers In-Use,
       (R) Default Route Exists, (V) Linked VLAN is UP.
```

Display detailed information for a specific probe:

```
show service-probe <instance_id> detail
```

Example output (version 33.6.1 and later):

```
================================================================================
ID                    : 1
VLAN                  : SpBlue (UP)
In-Use MAC            : 0a:11:88:fe:ec:36
Default Route Exists  : True
Dynamic               : False
Cfg IP                : 150.150.1.99/24
Cfg Gateway           : 150.150.1.4
Cfg DNS               : 150.150.1.4
In-Use IP             : 150.150.1.99/24

In-Use Resource   Server              Status      Query Time
----------------- ------------------- ----------- -------------------
Gateway           150.150.1.4         Reachable   02-03-2026 12:37:15
Primary DNS       150.150.1.4         Success     02-03-2026 12:34:29
```

Display Service Probe cgroup resource usage:

```
# show proc group service-probe
SvcProbe:
    Number of processes                   :   0
    CPU
        Limit                             :   5   %
        Current utilization               :   0.0 %
        Maximum utilization               :   0.0 %
```

```
    Memory
        Upper Limit                     :   5   %
        Current utilization             :   0.0 %
        Maximum utilization             :   0.0 %
```

Configuration Examples

The following configuration examples describe various methods of configuring Service Probe.

Example: Verifying DHCP and DNS on a User VLAN

This example creates a probe on the `corp` VLAN using DHCP, then verifies that clients on that VLAN can reach DNS.

```
# Create the probe — it will attempt DHCP immediately
create service-probe 1 vlan corp

# Confirm the probe received an address and the VLAN is up
show service-probe 1 detail

# Test DNS resolution against all configured servers
run service-probe 1 query dns www.extremenetworks.com

# Narrow it down to the primary server if a failure is seen
run service-probe 1 query dns primary www.extremenetworks.com
```

Example: Static IP Probe with Gateway and DNS Checks

This example creates a probe with a fixed IP address and validates both gateway reachability and DNS resolution — useful when you want repeatable results independent of DHCP.

```
# create a static probe
create service-probe 18 vlan corp100 ipaddress 192.168.200.93/24 192.168.200.10

# Add up to three DNS servers
configure service-probe 18 add name-server 1.1.1.1 8.8.4.4 208.67.222.222

# Verify the gateway is responding to ARP
run service-probe 18 query gateway

# Verify DNS resolution
run service-probe 18 query dns extreme.com

# Review full probe detail, including cached query results
show service-probe 18 detail
```

Example: Running a Traceroute via Shell

This example uses shell execution (version 33.6.1 and later) to run a traceroute from inside the probe's network namespace, revealing the actual path traffic takes from the client's perspective.

```
# Create a static probe on the uplink VLAN
create service-probe 1 vlan uplink ipaddress 172.16.1.99/24 172.16.1.1
```

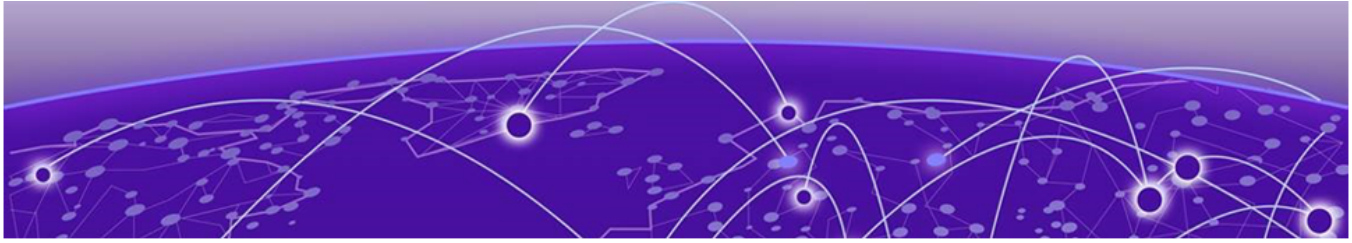
```
# Launch a shell command in the probe's context
run service-probe 1 shell
```

At the prompt, enter the command:

```
traceroute 11.100.100.1
```

Expected output:

```
Exit Status: 0
...output follows:
traceroute to 11.100.100.1 (11.100.100.1), 30 hops max, 46 byte packets
 1  172.16.1.1 (172.16.1.1)    0.547 ms  0.343 ms  0.321 ms
 2  11.100.100.1 (11.100.100.1) 0.287 ms  0.271 ms  0.247 ms
```



Software Upgrade and Boot Options

ExtremeXOS Upgrade Process on page 1697
Installing a Modular Software Package on page 1711
Configuration Changes on page 1712
Using TFTP to Upload the Configuration on page 1719
Using TFTP to Download the Configuration on page 1720
Synchronizing Nodes on page 1721
Upgrading the Bootrom on page 1722
Upgrading the Firmware on page 1723
Displaying the BootROM and Firmware Versions on page 1724

This chapter provides instructions for upgrading ExtremeXOS software and firmware, including how to download and install a new image, perform a hitless upgrade, upload and download the configuration using TFTP, and synchronize nodes.

To upgrade your ExtremeXOS images and modules, start with ExtremeXOS Upgrade Process on page 1697.

To install modules on top of an existing ExtremeXOS image that will not be upgraded, see Installing a Modular Software Package on page 1711.

ExtremeXOS Upgrade Process

Upgrading your ExtremeXOS image and modules uses the following process:

1. Backup all configuration files (see Creating a Backup Configuration File on page 1700).
2. Download ExtremeXOS and module package images from the Extreme Networks support portal (see Downloading a New Image on page 1704).

   Note
   Beginning with ExtremeXOS 16.1, there are two methods available for the upgrade process:
   - `download url` *`filename`* - provides a means of downloading multiple files at the same time over TFTP, HTTP, or anonymouns FTP.
   - `download image` *`filename`* - downloads a single .xos or .xmod file from a TFTP server.

3. Place the image files on a server that your switch can locate (see Installing a Core Image on page 1707).

4. Discover your inactive partition (see Finding the Inactive Partition on page 1706).
5. Use the `download image` command to install each image on the inactive partition (see Installing a Core Image on page 1707).
6. Reboot the switch (see Reboot Options on page 1708).

Upgrade Example

We recommend upgrading software images and any modular packages at the same time. The following example shows an ExtremeSwitching 5720 image onie-32.2.0.416.x86_64.xos being installed before rebooting the switch:

```
# show switch

SysName:          5720-48MW-SwitchEngine
SysLocation:
SysContact:       https://www.extremenetworks.com/support/
System MAC:       F4:CE:48:04:90:00
System Type:      5720-48MW-SwitchEngine

SysHealth check:  Enabled (Normal)
Recovery Mode:    All
System Watchdog:  Enabled
Trusted Delivery: Boot Image Verified

Current Time:     Wed Sep 14 19:58:24 2022
Timezone:         [Auto DST Disabled] GMT Offset: 0 minutes, name is UTC.
Boot Time:        Wed Sep 14 19:56:31 2022
Boot Count:       108
Next Reboot:      None scheduled
System UpTime:    1 minute 52 seconds

Current State:    OPERATIONAL
Image Selected:   primary
Image Booted:     primary
Primary ver:      32.2.0.414
Secondary ver:    32.2.0.414

Config Selected:  primary.cfg
Config Booted:    primary.cfg
Config Automatic: NONE (Disabled)

primary.cfg       Created by Extreme Networks Switch Engine version 32.2.0.414
                  1031871 bytes saved on Tue Sep 13 21:20:10 2022


LAA MAC:          Locally Administered MAC Address Disabled

# download image 10.127.4.103 onie-32.2.0.416.x86_64.xos secondary
Do you want to install image after downloading? (y - yes, n - no, <cr> - cancel) Yes

Downloading to
Switch.....................................................................................
.....................................................................................
.....................................................................................
...................
Installing to secondary partition!

Installing to Switch.............................................
This xos package does not include a switch diagnostics image and no diagnostics image is
installed on the switch.  Please download and install the diagnostics xmod with the
download image command.
```

```
Image installed successfully
This image will be used only after rebooting the switch!
* (CIT_32.2.0.414) 5720-48MW-SwitchEngine.8 # reboot
Do you want to save configuration changes to currently selected configuration
file (primary.cfg) and reboot?
(y - save and reboot, n - reboot without save, <cr> - cancel command)

# reboot
Do you want to save configuration changes to currently selected configuration
file (primary.cfg) and reboot?
(y - save and reboot, n - reboot without save, <cr> - cancel command)
```

The following example shows an ExtremeSwitching 5520 image summit_arm-32.2.0.416.xos being installed before rebooting the switch:

```
# show switch

SysName:          5520-48W-SwitchEngine
SysLocation:
SysContact:       https://www.extremenetworks.com/support/
System MAC:       00:04:96:EC:A4:00
System Type:      5520-48W-SwitchEngine

SysHealth check:  Enabled (Normal)
Recovery Mode:    All
System Watchdog:  Enabled
Trusted Delivery: Boot Image Verified

Current Time:     Wed Sep 14 20:04:40 2022
Timezone:         [Auto DST Disabled] GMT Offset: 0 minutes, name is UTC.
Boot Time:        Wed Sep 14 19:56:10 2022
Boot Count:       62
Next Reboot:      None scheduled
System UpTime:    8 minutes 30 seconds

Current State:    OPERATIONAL
Image Selected:   primary
Image Booted:     primary
Primary ver:      32.2.0.411
Secondary ver:    32.2.0.414

Config Selected:  primary.cfg
Config Booted:    primary.cfg
Config Automatic: NONE (Disabled)

primary.cfg       Created by Extreme Networks Switch Engine version 32.2.0.414
                  1031871 bytes saved on Tue Sep 13 21:20:10 2022


LAA MAC:          Locally Administered MAC Address Disabled

# show version detail

Switch          : 800994-00-03 2008G-01290 Rev 03 BootROM: 2.2.1.11   IMG: 32.2.0.411
                  CPLD: 1.0.24 UC: 1.0.12
5520-VIM-4YE-1  : 800999-00-03 SB102011G-00319 Rev 03

Image   : Extreme Networks Switch Engine version 32.2.0.411 by release-manager
          on Mon 12 Sep 2022 01:36:37 AM UTC
BootROM : Default 2.2.1.11  Alternate 2.2.1.11
Diagnostics :
Certified Version : EXOS Linux 5.4.183, Extreme Networks FIPS Object Module 2.0.16a
```

```
Build Tools Version : exos-arm64-sdk-3.1.14.1.0

# download image 10.127.4.103 summit_arm-32.2.0.416.xos secondary

Do you want to install image after downloading? (y - yes, n - no, <cr> - cancel) Yes

Downloading to
Switch.....................................................................................
.....................................................................................
.....................................................
Installing to secondary partition!

Installing to
Switch.....................................................................................
.....................................................................................
.....................................................................
Image installed successfully
Bootrom versions are current.
This image will be used only after rebooting the switch!

# reboot

Do you want to save configuration changes to currently selected configuration
file (primary.cfg) and reboot?
(y - save and reboot, n - reboot without save, <cr> - cancel command)
```

Creating a Backup Configuration File

Before upgrading ExtremeXOS on a switch, it is highly recommended that you create a backup copy of the configuration. Creating a copy of the "Config Booted" configuration file, with the version number of the ExtremeXOS image included in the name of the backup copy, is the simplest method.

To do this:

1. Save the configuration to the database so the switch can reapply the configuration after the switch reboot using the `save` command.
2. Enter `y` at the prompt to save your changes.
3. Execute the `show switch` command to view the "Config Booted" file. The following output is displayed:

```
# show switch
SysName:          XXXXX
SysLocation:
SysContact:       support@extremenetworks.com, +1 888 257 3000
System MAC:       00:04:96:51:FE:E2
System Type:      XXXXXX

SysHealth check:  Enabled (Normal)
Recovery Mode:    All
System Watchdog:  Enabled

Current Time:     Thu Sep  4 00:57:18 2014
Timezone:         [Auto DST Disabled] GMT Offset: 0 minutes, name is UTC.
Boot Time:        Wed Sep  3 20:07:11 2014
Boot Count:       402
Next Reboot:      None scheduled
System UpTime:    4 hours 50 minutes 7 seconds

Current State:    OPERATIONAL
```

```
Image Selected:   primary
Image Booted:     primary
Primary ver:      15.7.1.4
Secondary ver:    15.7.1.5

Config Selected:  ssh-privatekey.cfg
Config Booted:    ssh-privatekey.cfg
                  ssh-privatekey.cfg Created by ExtremeXOS version
15.7.1.5
            219131 bytes saved on Mon Jul 14 23:03:08 2014
```

4. Using the ExtremeXOS version shown in the "Config Booted:" line, enter the "copy" command as shown:

   `cp config selected config-booted-exos_ver`

   For example: `cp ssh-privatekey.cfg ssh-privatekey-15_7_1_5`

   Note
   For more information about version strings, see Understanding the Image Version String on page 1709.

For more information about ExtremeXOS configuration files, see Configuration Changes on page 1712.

Download URL Method

ExtremeXOS 21.1 provides an enhancement to the download CLI command so that it now accepts a URL as the name of the file to download.

Note
This feature is not available unless you are currently running ExtremeXOS 21.1 or later.

URL protocols can be TFTP, HTTP, or anonymous FTP. The format of a URL is:

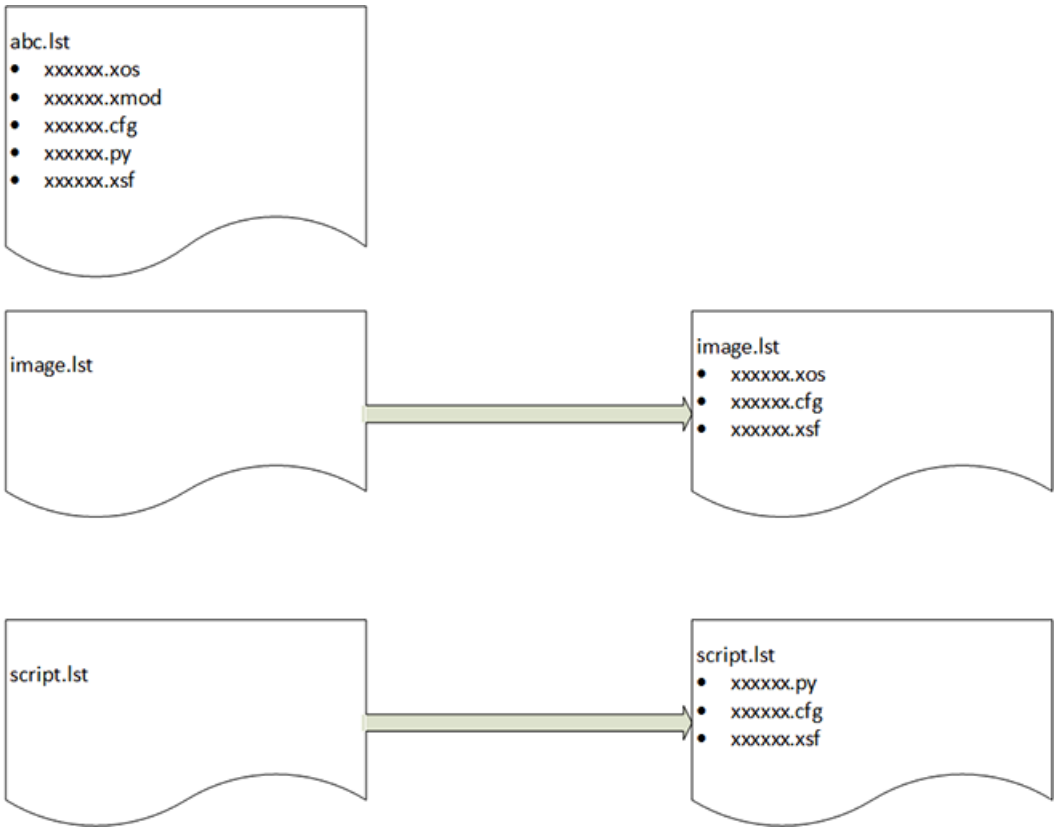
- http://10.10.10.1/filename.xos
- tftp://10.10.10.1/filename.xos
- ftp://10.10.10.1/filename.xmod

In addition to accepting a URL that ends in .xos or .xmod, the URL filename can now end in .lst.

A .lst file contains filenames at the same location as the .lst file URL and will be downloaded, and then installed one after the other. The .lst file method can define bundles of downloads for:

- Large image file sizes
- Files ending in '.cfg', '.xsf', '.pol', '.xlic', '.py', '.ssh'
- Any additional .lst files so that you can create lists of lists
- Any bundling that makes it easier to download with a single command

The .lst file allows you to download .xos, .xmods, and other files at the same time. Once downloaded as a list, simply reboot and the files become active. This feature allows for better organization of downloads across switches.



All of the filenames in any .lst must be available at the same URL path provided in the download command.

Any .xos and .xmod files are installed to the inactive partition.

For any other file types, the /usr/local/cfg directory is used.

See the following for an example of .lst files

Example

An HTTP server on 10.68.9.7 for directory 16.1/xxxx/xxxx/release:

This example shows how the .lst file can contain filenames ending in .lst to get a list of lists (of lists etc...)

cat big.lst – big.lst contains other list file names

xos.lst

xmod.lst

script.lst

cat xos.lst – xos.lst contains an ExtremeXOS image

summitX-16.1.0.18.xos

cat xmod.lst – xmod.lst contains a number of .xmod filenames

summitX-16.1.0.18-debug.xmod

summitX-16.1.0.18-LegacyCLI.xmod

summitX-16.1.0.18-reachnxt-1.8.1.8.xmod

summitX-16.1.0.18-techSupport.xmod

cat script.lst – script.lst contains a number of python scripts the user wants to download to a switch

jsonrpc.py

jsontest.py

otst.py

ping.py

readvr.py

A single download command downloads all of the above files. Example uses big.lst:

```
# download url http://10.68.9.7/big.lst
http://10.68.9.7/xos.lst
Downloading http://10.68.9.7/summitX-16.1.0.18.xos

Downloading to
Switch.....................................................................................
.................................................
Installing to primary partition!

Installing to
Switch.....................................................................................
.....................................................................................
.....................
Image installed successfully
This image will be used only after rebooting the switch!

Image installed successfully
http://10.68.9.7:8080/xmod.lst
Downloading http://10.68.9.7/summitX-16.1.0.18-debug.xmod

Downloading to Switch.....
Installing to primary partition!

Installing to Switch.........................................
Image installed successfully
Downloading http://10.68.9.7/summitX-16.1.0.18-LegacyCLI.xmod

Downloading to Switch..
```

```
Installing to primary partition!

Installing to Switch.......................
Legacy CLI framework was Successfully Installed !!!

Image installed successfully
Downloading http://10.68.9.7/summitX-16.1.0.18-reachnxt-1.8.1.8.xmod

Downloading to Switch...
Installing to primary partition!

Installing to Switch....
Image installed successfully
Downloading http://10.68.9.7/summitX-16.1.0.18-techSupport.xmod

Downloading to Switch..
Installing to primary partition!

Installing to Switch..
Image installed successfully
http://10.68.9.7/script.lst
http://10.68.9.7/jsonrpc.py
http://10.68.9.7/jsontest.py
http://10.68.9.7/otst.py
http://10.68.9.7/ping.py
http://10.68.9.7/readvr.py
```

Downloading a New Image

The ExtremeXOS core image (.xos) file contains the executable code that runs on the switch and is preinstalled at the factory. An ExtremeXOS module image (.xmod) supplements the core image. Note that the version number of the core image and the module must match. As new versions of this image are released, you should upgrade both the software and modular packages running on your system.

To find and install image files:

1. Point your browser to http://esupport.extremenetworks.com and log in with your eSupport account.

   If you do not have an eSupport account, you can request one with the **Request Web Login** link.

   Note
   You can also download release notes from the eSupport site. The *ExtremeXOS Release Notes* will always have the most current download and install instructions.

2. Find the ExtremeXOS image and all modules you wish to upgrade. Later you will install all files on the image at the same time.
3. Save the files to a TFTP server on the network. You can also store them on a USB 2.0 device.
4. When you have all images downloaded and saved, proceed to Finding the Inactive Partition on page 1706.

Image Integrity Checking

This feature adds digital signature verification in ExtremeXOS image and XMOD modules. Image integrity is checked against the digital signature before actual installation.

Digital Signature

Prior to ExtremeXOS 16.1, the ExtremeXOS image or XMOD modules downloaded to the switch were only protected by CRC check. This is sufficient for checking corrupted image. However, it cannot prevent malicious attacks. For example, an attacker can de-package the image, replace certain fields, recompute CRC and then repackage the image.

Digital signature is commonly used to demonstrate of the authenticity of a digital message, in this case, the image downloaded to the switch. Only images with digital signature validated on the switch can be installed. Otherwise, the installation should be aborted.

This features uses the Public Key Infrastructure (PKI) approach. Specifically, with the RSA algorithm, two keys, the private key and the public key are generated using openssl utility. The ExtremeXOS image or XMOD module is digitally signed with the private key. The public key is installed on the swtich in the format of a X.509 certificate, which is verified before being used.

When building the image, the signature is computed for the ExtremeXOS image or XMOD module, then included in the final image provided to the customer. On the switch, during downloading process, the signature is verified against the image using the public key previously installed.

In order to deliver the public key to the customer securely, it is also digitally signed and it is distributed in the format of a X.509 certificate. In order to do so, another set of keys are generated to sign this certificate. A self-signed root certificate is also installed on the switch to verify the certifcate containing the image signing public key.

All these keys and certificates are generated offline and the private keys should be stored safely.

Transition from an image without signature to one with needs two steps, the first step is to download the ExtremeXOS image and install the public key certificates. At this time, the signature cannot be verified because there is no key to validate the image. But after the first installation, all subsequent downloaded images can be validated using the installed key.

The certificates are only included in the ExtremeXOS image; XMOD modules do not need to include certificates.

Downgrading from an ExtremeXOS version supporting digital signature to one that does not is allowed. No special handling is needed. A warning message is printed on the console to remind the user that the image is not digitally singed. The user does have the choice to either proceed with downgrading or not.

Hash Verification

Currently, each ExtremeXOS image or XMOD is posted along side with an *MD5 (Message-Digest algorithm 5)* hash checksum, which can be verified using any offline tools. As of ExtremeXOS 16.1, an enhancement now provides a stronger SHA256 hash checksum is generated for you to verify offline.

Finding the Inactive Partition

A switch can store up to two core images: an active and inactive. When downloading a new image, you must select on which partition to install the new image. You must install the software image to the inactive partition, and must specify that partition while downloading the image to the switch.

To find the inactive partition:

1. From the CLI, enter the command `show switch` (or `show slot detail` on SummitStack).

```
# show switch
SysName:          XXXXXXX
SysLocation:
SysContact:       support@extremenetworks.com, +1 888 257 3000
System MAC:       00:04:96:51:FE:E2
System Type:      XXXXXXX

SysHealth check:  Enabled (Normal)
Recovery Mode:    All
System Watchdog:  Enabled

Current Time:     Thu Sep  4 00:57:18 2014
Timezone:         [Auto DST Disabled] GMT Offset: 0 minutes, name is UTC.
Boot Time:        Wed Sep  3 20:07:11 2014
Boot Count:       402
Next Reboot:      None scheduled
System UpTime:    4 hours 50 minutes 7 seconds

Current State:    OPERATIONAL
Image Selected:   primary
Image Booted:     primary
Primary ver:      15.7.1.4
Secondary ver:    15.7.1.5

Config Selected:  ssh-privatekey.cfg
Config Booted:    ssh-privatekey.cfg
                  ssh-privatekey.cfg Created by ExtremeXOS version
15.7.1.5
            219131 bytes saved on Mon Jul 14 23:03:08 2014
```

2. Locate the `Image Booted:` line. If it indicates `primary`, you will know to install the images on the secondary partition. If it indicates `secondary`, you will know to install the images on the primary partition.

   For help with versions and image filenames, refer to the following:

   - Understanding the Image Version String on page 1709
   - Image Types and Image Filename Prefixes
   - Software Signatures on page 1710

When you know the inactive partition, proceed to Installing a Core Image on page 1707.

Installing a Core Image

Once .xos and .xmod files are stored on a network server (see Downloading a New Image on page 1704), download the image to the switch using the following procedure:

1. Use a login session on the master node of the SummitStack to verify which connects to your server. Use one of the following ping commands to confirm which virtual router reaches your server:

   ```
   ping vr vr-Mgmt host
   ping vr vr-Default host
   ```

   At least one of these commands must successfully reach your server for you to download the image. After verifying the virtual router that reaches your server, specify that virtual router when you download the image.
2. Download the new image to the switch using the command:

   ```
   download [url url {vr vrname} | image [active | inactive] [[hostname | ipaddress] filename {{vr} vrname} {block-size block_size}] {partition} {install {reboot}}
   ```

   Note
   If you configure the switch to write core dump (debug) files to the internal memory card and attempt to download a new software image, you might have insufficient space to complete the image download. If this occurs, you will need to move or delete the core dump files from the internal memory. You will be asked during the download process if you want to remove these files. For more information, see Understanding Core Dump Messages on page 1710.
3. Before the download begins, the switch asks if you want to install the image immediately after the download is finished. Enter `y` to install the image after download.

   If you enter `n` to install the image at a later time, the image is still downloaded and saved to the switch, but you must use the `install image` command when ready to install the image.
4. Install .xmod images to the inactive partition using the same command shown above.
5. After installing the .xos and all .xmod iamges, reboot the switch (see Reboot Options on page 1708).

For information about installing a new BootROM image on an ExtremeSwitching series switch, see Upgrading the Bootrom on page 192.

Installing a Core Image with NMS

You can either load the new image to the NMS server on your network or configure it to automatically poll and download newly released images to the server. Follow the instructions described in the NMS documentation to appropriately configure the server for your network environment.

For more information about installing the NMS client and server, configuring it, and the platforms the NMS supports, refer to the NMS documentation that comes with

the product or the documentation available from the Extreme Networks website at www.extremenetworks.com/documentation/.

Copying an Image from the Active to Inactive Partition

To copy the image on the active partition to the inactive partition, use the following command:

```
install image inactive {slot slot}
```

Reboot Options

Reboot the Switch

- To reboot the switch immediately, use the following command
  `reboot`

  Note
  The reboot occurs immediately and any previously schedule reboots are cancelled.

  On SummitStack, the `reboot` command reboots the *active* topology and can be run from the master node only.
- To schedule a time to reboot the switch, use the command:
  `reboot {[time mon day year hour min sec] | cancel} {slot slot-number} | node-address node-address | stack-topology {as-standby} | all} | rolling}`

  The options use the following format:
  - **date** — in mm dd yyy format
  - **time** — 24-hour clock in hh mm ss format

  Note
  When you configure a timed reboot of the switch, you can use the `show switch` command to see the scheduled time.

- To reboot the entire stack topology, use the command:
  `reboot stack-topology {as-standby}`

  Note
  This command can be run on any node and can be used to eliminate the dual master condition manually.

- To cancel a previously scheduled reboot, use the **cancel** option of the `reboot` command.

Reboot a Node in a SummitStack

To reboot a single node in the SummitStack, use the command:

```
reboot {[time mon day year hour min sec] | cancel} {slot slot-number}
| node-address node-address | stack-topology {as-standby} | all} |
rolling}
```

The `reboot slot` command works only on the active master node for other slots, and on an active non-master node for its own slot.

The reboot `stack-topology {as-standby}` command can also be used to reboot a single node by specifying the MAC address of the node.

The `reboot node-address` command reboots the specified node from any node.

Understanding the Image Version String

The image version string contains build information for each version of ExtremeXOS.

You can use the `show version` and `show switch` commands to display the ExtremeXOS version running on your switch. The output will be structured as one of the following:

- ExtremeXOS Version <major>.<minor>.<sustaining>.<build>

  In the case of a patch release, the version structure is ExtremeXOS Version <major>.<minor>.<sustaining>.<build>-patchX-Y where X is the patch type and Y is the build number for the patch.
- <major>.<minor>.<sustaining>.<build>patchX-Y

Table 149 describes the image version fields.

Table 149: Image Version Fields

Field	Description
major	Specifies the ExtremeXOS major version number.
minor	Specifies the ExtremeXOS minor version number.
sustaining	Specifies the ExtremeXOS sustaining version number.
build	Specifies the ExtremeXOS build number.
patch	Identifies a specific patch release.

The `show version` command also displays information about the firmware (BootROM) images running on the switch. For more information, see Displaying the BootROM and Firmware Versions on page 1724.

Image Types and Image Filename Prefixes

The software image file can be an .xos file, which contains a Switch Engine core image, or an .xmod file, which contains a Switch Engine modular software package.

Filename Prefixes

You can identify the appropriate image or module for your platform based on the filename prefix of the image.

Table 150: Switch Engine Image Types (Prefixes)

Switches	Image File Type (Prefix)
4120, 5120	`rzg2` Example: `rzg2-33.1.1.x.xos`
4220, 5320, 5420, 5520	`summit_arm` Example: `summit_arm-31.3.1.x.xos`
5720, 7520, 7720	`onie` Example: `onie-32.1.1.6.x86_64.xos`

For additional installation requirements see the sections, Installing a Core Image on page 1707 and Installing a Modular Software Package on page 1711.

Software Signatures

Each ExtremeXOS image contains a unique signature.

The BootROM checks for signature compatibility and denies an incompatible software upgrade. In addition, the software checks both the installed BootROM and software and also denies an incompatible upgrade.

Understanding Core Dump Messages

If you configure the switch to write core dump (debug) files to internal memory (`/usr/local/tmp`) and attempt to download a new software image, you might have insufficient space to complete the image download.

If this occurs, move or delete the core dump files from the internal memory. For example, if the switch supports a USB 2.0 storage device and space is available, transfer the files to the storage device. On switches without removable storage devices, transfer the files from internal memory to a TFTP server. This frees up space on the internal memory card while keeping the core dump files.

The switch displays a message similar to the following and prompts you to take action:

```
Core dumps are present in internal-memory and must be removed before this download can
continue.
(Please refer to documentation for the “configure debug core-dumps” command for
additional information)
Do you want to continue with download and remove existing core dumps? (y/n)
```

Type `y` to remove the core dump files and download the new software image. Type `n` to cancel this action and transfer the files before downloading the image.

For information about configuring and sending core dump information, see the `configure debug core-dumps [ off | directory_path]` and `save debug tracefiles directory_path` commands.

Installing a Modular Software Package

In addition to the functionality available in the ExtremeXOS core image, you can add functionality to your switch by installing modular software packages or feature packs. A complete listing of these feature packs can be found with their availability requirements in the Switch Engine v33.6.1 Licensing Guide document.

Modular software packages are contained in files named with the file extension .xmod, while the core images use the file extension .xos.

Modular software packages are built at the same time as core images and are designed to work in concert with the core image, so the <major>.<minor>.<patch> field of a modular software package must match the <major>.<minor>.<patch> field of the core image that it will be running with. For example, the modular software package for tech support named:

```
summitX-21.1.1.4-techSupport.xmod
```

Can run only with the core image named:

```
summitX-21.1.1.4.xos
```

You can install a modular software package on the active or inactive partition. You would install on the active partition if you want to add the package functionality to the currently running core image without having to reboot the switch. You would install on the inactive partition if you want the functionality available after a switch reboot.

To install the package on the inactive partition, you use the same process that you use to install a new core image. Follow the process described in Installing a Core Image on page 1707.

Activate the installed modular software package either by rebooting the switch or by issuing the following command: `run update`

You can uninstall packages by issuing the following command:

`uninstall image fname partition {msm slotid} {reboot}`

Note

Do not terminate a process that was installed since the last reboot unless you have saved your configuration. If you have installed a software module and you terminate the newly installed process without saving your configuration, your module may not be loaded when you attempt to restart the process with the `start process` command.

Upgrading a Modular Software Package

When Extreme Networks introduces a new core software image, a new feature pack or modular software package is also available. If you have a software module installed and upgrade to a new core image, you need to upgrade to the corresponding feature pack at the same time (see ExtremeXOS Upgrade Process on page 1697).

If you wish to upgrade your existing module software package *without* upgrading your ExtremeXOS image, two methods are available. Regardless of which method you choose, you must terminate and restart the processes associated with the software module.

1. Method One
   a. Terminate the processes associated with the software module using one of the following commands:
      `terminate process` *name* [**forceful** | **graceful**] (modular switches)
      `terminate process` *name* [**forceful** | **graceful**] {**slot** *slot*} (SummitStack)
   b. Download the software module from your TFTP server, compact flash card, or USB 2.0 storage device using the following command:
      `download` [**url** *url* {**vr** *vrname*} | `image` [**active** | **inactive**] [[*hostname* | *ipaddress*] *filename* {{**vr**} *vrname*} {**block-size** *block_size*}] {*partition*} {**install** {**reboot**}}
   c. Activate the installed modular package, if installed on the active partition, using the following command: `run update`
   d. Restart the processes associated with the software module using one of the following commands:
      `start process` *name* {**msm** *slot*} (modular switches)
      `start process` *name* {**slot** *slot*} (SummitStack)
2. Method Two
   a. Download the software module from your TFTP server, compact flash card, or USB 2.0 storage device using the following command:
      `download` [**url** *url* {**vr** *vrname*} | `image` [**active** | **inactive**] [[*hostname* | *ipaddress*] *filename* {{**vr**} *vrname*} {**block-size** *block_size*}] {*partition*} {**install** {**reboot**}}
   b. Activate the installed modular package, if installed on the active partition, using the following command: `run update`
   c. Terminate and restart the processes associated with the software module using one of the following commands:
      `restart process` [**class** *cname* | *name* {**msm** *slot*}] (modular switches)
      `restart process` [class *cname* | *name* {**slot** *slot*}] (SummitStack)

Configuration Changes

Image Configuration Overview

The configuration is the customized set of parameters that you have selected to run on the switch. As you make configuration changes, the new settings are stored in run-time

memory. Settings that are stored in run-time memory are not retained by the switch when the switch is rebooted. To retain the settings and have them loaded when you reboot the switch, you must save the configuration to nonvolatile storage.

The switch can store multiple user-defined configuration files, each with its own filename. By default, the switch has two prenamed configurations: a primary and a secondary configuration. When you save configuration changes, you can select to which configuration you want the changes saved or you can save the changes to a new configuration file. If you do not specify a filename, the changes are saved to the configuration file currently in use. Or if you have never saved any configurations, you are asked to save your changes to the primary configuration.

Note
Configuration files have a .cfg file extension. When you enter the name of the file in the CLI, the system automatically adds the .cfg file extension.

If you have made a mistake or you must revert to the configuration as it was before you started making changes, you can tell the switch to use the backup configuration on the next reboot.

Each filename must be unique and can be up to 32 characters long. Filenames are also case sensitive. For information about filename restrictions, see the specific command in the Switch Engine v33.6.1 Command References .

- First, save the configuration using the command:

  `save configuration {`**`primary`** `|` **`secondary`** `|` *`existing-config`* `|` *`new-config`*`}`

  Where the options are:

  - **`primary`** — Specifies the primary saved configuration
  - **`secondary`** — Specifies the secondary saved configuration
  - **`existing-config`** — Specifies an existing user-defined configuration (displays a list of available user-defined configuration files)
  - **`new-config`**—Specifies a new user-defined configuration
- You are prompted to save the changes. Enter `y` to save the changes or enter `n` to cancel the process.
- Next, use the configuration command:

  `use configuration` *`file-name`*

  Where the options are:

  - **`primary`** — Specifies the primary saved configuration
  - **`secondary`** — Specifies the secondary saved configuration
  - **`file_name`** — Specifies an existing user-defined configuration (displays a list of available user-defined configuration files)

The configuration takes effect on the next reboot.

Note
When a switch boots up after executing this command, examine the log to determine if a subsequent save and a reboot of the switch or slot is necessary for the configuration to be correctly applied.

Note
If the switch is rebooted while in the middle of saving a configuration, the switch boots to factory default settings if the previously saved configuration file is overwritten. The configuration that is not in the process of being saved is unaffected.

Automatic Saving of Configuration Files

The currently running switch configuration can be configured to be saved at automatic intervals. This feature is turned off by default, but can be enabled and configured to save at specified time intervals and to a designated file.

To configure automatic configuration saving, use the command `save configuration **automatic** {**every** *minutes* {**primary** | **secondary** | *existing-config* | *new-config*} | **never**}` .

Note
Automatic save of the running configuration on system reboot, irrespective of the interval between saves is handled for only immediate reboot, not for scheduled reboots or rebooting through other technique,s such as *SNMP (Simple Network Management Protocol)*.

View a Configuration

- To view the current configuration, enter the command: on the switch.
  `show configuration {*module-name*} {**detail**}`

  You can also view just that portion of the configuration that applies to a particular module (for example, *SNMP*) by using the *module-name* parameter.

  Beginning with ExtremeXOS 12.1, when you specify show configuration only, the switch displays configuration information for each of the switch modules excluding the default data.

  You can send output from the `show configuration {*module-name*} {**detail**}` command to the Extreme Networks Technical Support for problem-solving purposes. The output maintains the command line interface (CLI) format of the current configuration on the switch.

Comparing Configurations

You can view the differences between switch configuration files or a configuration file with the running configuration using the command:

```
show configuration difference { from-config-file {to-config-file} }
{module-name} {detail} {context lines}
```

When using the `save configuration automatic {every minutes {primary | secondary | existing-config | new-config} | never}` command this can be especially helpful for comparing what changed between the running (or "startup") configuration and the most recently autosaved configuration.

Restore Factory Defaults

The following two procedures restore the switch to factory defaults and reboot the switch. Use the first procedure when you can log in to the switch and issue a CLI command and the second when you cannot.

- When you can log in to the switch, use the following command:

  ```
  unconfigure switch
  ```

  This command resets most of the configuration, with the exception of user-configured user accounts and passwords, the date, and the time. On SummitStack, the command also preserves stacking-specific parameters so the stack can be formed after reboot.
- Unset the currently selected configuration image, reset all switch parameters, and reboot the switch .

  ```
  unconfigure switch all
  ```
- If you cannot log in because the switch is in a continuous booting loop, use the following procedure:
  a. Reboot the switch while pressing the space bar. This puts the switch in Bootstrap mode.
  b. From the Bootstrap prompt, type `boot` and then press **[Enter]** and press the spacebar. This puts the switch in BootROM mode.
  c. From the BootROM prompt, type `config none`.

     The following appears:
     ```
     Configuration selected: none
     ```
  d. From the BootROM prompt, enter `reboot`. The switch reboots and restores the factory defaults.

Uploading ASCII-Formatted Configuration Files

You can upload your current configuration in ASCII format to a TFTP server.

The uploaded ASCII file retains the CLI format and allows you do the following:

- View and modify the configuration using a text editor, and later download a copy of the file to the same switch or to one or more different switches.

- Send a copy of the configuration file to Extreme Networks Technical Support for problem-solving purposes.

Summary of Tasks

The following summary describes only the CLI involved to transfer the configuration and load it on the switch; it is assumed that you know how to modify the configuration file with a text editor. As previously described, to use these commands, use the .xsf file extension, as these steps are not applicable to configurations that use the .cfg file extension.

To work with an ASCII-formatted configuration file, complete the following tasks:

1. Upload the configuration to a network TFTP server using the following command:

   `upload configuration [hostname | ipaddress] filename {vr vr-name} {block-size block_size}`

   After the configuration file is on the TFTP server, use a text editor to enter the desired changes, and rename the file if necessary so it has the .xsf extension.
2. Download the configuration from the TFTP server to the switch using one of the `tftp` and `tftp get` commands.
3. Verify the configuration file is on the switch using the `ls` command
4. Load and restore the new configuration file on the switch using the following command:

   `load script filename {arg1} {arg2} ... {arg9}`
5. Save the configuration to the configuration database so the switch can reapply the configuration after switch reboot using the following command:

   `save configuration {primary | secondary | existing-config | new-config}`

   When you save the configuration file, the switch automatically adds the .cfg file extension to the filename. This saves the ASCII configuration as an XML-based configuration file.

   Note
   Configuration files are forward compatible only and not backward compatible. That is, configuration files created in a newer release, such as ExtremeXOS 15.5, might contain commands that do not work properly in an older release, such as ExtremeXOS 15.3.

Upload the ASCII Configuration File To a TFTP Server

To upload the current switch configuration as an ASCII-based file to the TFTP server, use the `upload configuration` command and save the configuration with the .xsf file extension.

For example, to transfer the current switch configuration as an ASCII-based file named meg_upload_config1.xsf to the TFTP server with an IP address of 10.10.10.10, do the following:

```
upload configuration 10.10.10.10 meg_upload_config1.xsf
```

If you successfully upload the configuration to the TFTP server, the switch displays a message similar to the following:

```
Uploading meg_upload_config1.xsf to 10.10.10.10 ... done!
```

Download the ASCII Configuration File to the Switch

To download the configuration from the TFTP server to the switch, use the `tftp` or `tftp get` command.

For example, to retrieve the configuration file named meg-upload_config1.xsf from a TFTP server with an IP address of 10.10.10.10, you can use one of the following commands:

```
tftp 10.10.10.10 -g -r meg_upload_config1.xsf
tftp get 10.10.10.10 meg_upload_config1.xsf
```

If you successfully download the configuration to the switch, the switch displays a message similar to the following:

```
Downloading meg_upload_config1.xsf to switch... done!
```

Verify that the ASCII Configuration File is on the Switch

To confirm that the ASCII configuration file is on the switch, use the `ls` command. The file with an .xsf extension is the ASCII configuration.

The following sample output contains an ASCII configuration file:

```
-rw-r--r--    1 root     0           98362 Nov  2 13:53 Nov022005.cfg
-rw-r--r--    1 root     0              68 Oct 26 11:17 mcastgroup.pol
-rw-r--r--    1 root     0           21203 Dec 13 15:40 meg_upload_config1.xsf
-rw-r--r--    1 root     0          119521 Dec  6 14:35 primary.cfg
-rw-r--r--    1 root     0           96931 Nov 11 11:01 primary_11_11_05.cfg
-rw-r--r--    1 root     0           92692 Jul 19 16:42 secondary.cfg
```

Load the ASCII Configuration File

After downloading the configuration file, you must load the new configuration on the switch.

To load and restore the ASCII configuration file, use the command:

`load script` *`filename`* {**`arg1`**} {**`arg2`**} **`...`** {**`arg9`**}

After issuing this command, the ASCII configuration quickly scrolls across the screen.

The following is an example of the type of information displayed when loading the ASCII configuration file:

```
script.meg_upload_config1.xsf.389 # enable snmp access
script.meg_upload_config1.xsf.390 # enable snmp traps
script.meg_upload_config1.xsf.391 # configure mstp region purple
script.meg_upload_config1.xsf.392 # configure mstp revision 3
script.meg_upload_config1.xsf.393 # configure mstp format 0
script.meg_upload_config1.xsf.394 # create stpd s0
```

Instead of entering each command individually, the script runs and loads the CLI on the switch.

Save the Configuration

After you load the configuration, save it to the configuration database for use by the switch. This allows the switch to reapply the configuration after a switch reboot.

- To save the configuration, use the command;
  `save configuration {`**`primary`** | **`secondary`** | *`existing-config`* | *`new-config`*`}`

  When you save the configuration file, the switch automatically adds the .cfg file extension to the filename. This saves the ASCII configuration as an XML-based configuration file.

  You can use any name for the configuration.

  For example, after loading the file `meg_upload_config1.xsf`, you need to save it to the switch.

To save the configuration as configuration1.cfg, enter the command:

```
save configuration configuration1
```

Using Autoconfigure and Autoexecute Files

Two features allow you automatically execute scripts that can manage the switch configuration.

Autoconfigure:

Configuration commands placed in the default.xsf file are executed by the switch as it comes up and is unable to find its usual configuration file or if the switch is unconfigured or if the configuration file cannot be determined due to a corrupt NVRAM.

This returns the switch to some basic configuration. When default.xsf is executed, the `show switch` command shows default.xsf as the booted configuration file.

The default.xsf file can have any CLI commands as long as they are all executed within 500 seconds.

The script is aborted when the commands are not executed within that time. When the file is loaded, the results can be seen by executing the `show script output default` command.

Autoexecute:

Configuration commands placed in the autoexec.xsf file are executed after a switch loads its configuration.

The file is not executed when a default.xsf file has been executed. Use the file to execute commands after a switch is up and running and also to revert changes made to the configuration by UPM scripts that run persistent commands. The commands must be executed within 500 seconds or the script execution is aborted.

When an autoexec.xsf file is executed. the results can be seen by executing the `show script output autoexec` command.

Using TFTP to Upload the Configuration

You can upload the current configuration to a TFTP server on your network. Using TFTP, the uploaded configuration file retains your system configuration and is saved in XML format. This allows you to send a copy of the configuration file to Extreme Networks Technical Support for problem-solving purposes.

- To view your current switch configuration, use the `show configuration {module-name} {detail}` command available on your switch. Do not use a text editor to view or modify your XML-based switch configuration files.

  To view your current switch configuration in ASCII-format, see Uploading ASCII-Formatted Configuration Files on page 1715 for more information about uploading and downloading ASCII-formatted configuration files.

  For more information about TFTP, see Using the Trivial File Transfer Protocol on page 83.
- To upload the configuration from the switch to a TFTP server, you can use either the `tftp` or the `tftp put` command:

  ```
  tftp [ ip-address | host-name ] { -v vr_name } { -b block_size } [ -g | -p ] [ -l local-file { -r remote-file } | -r remote-file { -l localfile } ]
  ```

  Where the options are:
  - host-name — Specifies the host name of the TFTP server
  - ip-address — Specifies the IP address of the TFTP server
  - -p — Puts the specified file from the local host and copies it to the TFTP server
  - -l *local-file* — Specifies the name of the configuration file that you want to save to the TFTP server
  - -r *remote-file* — Specifies the name of the configuration file on the TFTP server

  ```
  tftp put [ ip-address | host-name] {vr vr_name} {block-size block_size}local-file { remote-file}
  ```

  Where the options are:
  - put —P uts the specified file from the local host and copies it to the TFTP server
  - host-name — Specifies the host name of the TFTP server
  - ip-address — Specifies the IP address of the TFTP server
  - *local-file* — Specifies the name of the configuration file that you want to save to the TFTP server
  - *remote-file* — Specifies the name of the configuration file on the TFTP server

If you upload a configuration file and see the following message:

```
Error: No such file or directory
```

Check to make sure that you entered the filename correctly, including the .cfg extension, and that you entered the correct host name or IP address for the TFTP server.

If your upload is successful, the switch displays a message similar to the following:

```
Uploading megtest1.cfg to TFTPhost ... done!
```

You can also upload the current configuration in ASCII format from the switch to a TFTP server on your network. For more information, see Uploading ASCII-Formatted Configuration Files on page 1715.

Using TFTP to Download the Configuration

You can download previously saved XML formatted XOS configuration files from a TFTP host to the switch to modify the switch configuration. Do not use a text editor to view or modify your switch configuration files; modify your switch configurations directly in the CLI.

To view your current switch configuration in ASCII-format, see Uploading ASCII-Formatted Configuration Files on page 1715 for more information about uploading and downloading ASCII-formatted configuration files.

For more information about TFTP, see Using the Trivial File Transfer Protocol on page 83.

Note
By default, if you transfer a file with a name that already exists on the system, the switch prompts you to overwrite the existing file. For more information, see the `tftp get` command in the Switch Engine v33.6.1 Command References .

- To download the configuration from a TFTP host to the switch, you can use either the tftp or the tftp get command:

  ```
  tftp [ ip-address | host-name ] { -v vr_name } { -b block_size } [ -g | -p ] [ -l local-file { -r remote-file } | -r remote-file { -l localfile } ]
  ```

  Where the options are:
  - **host-name** — Specifies the host name of the TFTP server
  - **ip-address** — Specifies the IP address of the TFTP server
  - **-p** — Puts the specified file from the local host and copies it to the TFTP server
  - **-l** *local-file* — Specifies the name of the configuration file that you want to save to the TFTP server
  - **-r** *remote-file* — Specifies the name of the configuration file on the TFTP server

  ```
  tftp get [ ip-address | host-name] { vr vr_name } { block-size block_size } remote-file local-file} {force-overwrite}
  ```

  Where the optons are:
  - **get** — Gets the specified file from the TFTP server and copies it to the local host
  - *host-name* — Is the host name of the TFTP server
  - *ip-address* — Is the IP address of the TFTP server
  - *remote_file* — Specifies the name of the configuration file that you want to retrieve from the TFTP server

- *local-file* — Specifies the name of the configuration file on the switch
- **force-overwrite** — Specifies the switch to automatically overwrite an existing file

If you download a configuration file and see the following message:

```
Error: Transfer timed out
```

Make sure that you entered the filename correctly, including the .cfg extension, and that you entered the correct host name or IP address for the TFTP server.

If your download is successful, the switch displays a message similar to the following:

```
Downloading megtest2.cfg to switch... done!
```

Configurations are downloaded and saved into the switch nonvolatile memory. The configuration is applied after you reboot the switch.

If the configuration currently running in the switch does not match the configuration that the switch used when it originally booted, an asterisk (*) appears before the command line prompt when using the CLI.

You can also download the current configuration in ASCII format from a TFTP server on your network to the switch. For more information, see Uploading ASCII-Formatted Configuration Files on page 1715.

Synchronizing Nodes

Before synchronizing nodes on a SummitStack, confirm that all nodes are active nodes and running ExtremeXOS 12.0 or later, so as to support the `synchronize` command.

On a SummitStack with redundancy, you can take the primary node configurations and images and replicate them on the backup node using the `save configuration` command.

Caution
During a synchronization on a SummitStack, the active stack briefly alternates between a ring and daisy-chain topology. When the primary node finishes replicating its configurations and images to the backup node, the full the stack ring is restored.

In addition to replicating the configuration settings and images, this command also replicates which configuration or image the node should use on subsequent reboots. This command does not replicate the run-time configuration. You must use the `save configuration {`**primary** | **secondary** | *existing-config* | *new-config*`}` command to store the run-time configuration first.

On a SummitStack, you can synchronize an active node in the stack with the primary or non-primary slots using the following command:

```
synchronize {slot slotid} {file_name}
```

Note
You cannot synchronize a master switch running a version earlier than ExtremeXOS 22.2 to a target node running ExtremeXOS 22.2 or later. The command fails and the following error message appears: `Error: the target slot's partitions are not compatible with the Master's for synchronize.`

Automatic Synchronization of Configuration Files

On a SummitStack where redundancy is in use, ExtremeXOS automatically synchronizes all of the configuration files from the primary node to the backup node if the switch detects that the backup node's configuration file contents are different from the primary node.

You cannot configure this behavior.

The switch deletes the old configuration files on the backup node only upon a successful file synchronization. If an error occurs, the switch does not delete the old configuration files on the backup node. For example, if you install a backup node that contains different configuration files from the primary node, the old configuration files are deleted after a successful bootup of the backup node.

To see a complete listing of the configuration files on your system, use the `ls` command.

For more detailed information, see Replicating Data Between Nodes on page 87.

Upgrading the Bootrom

The ExtremeSwitching series switches have a two-stage bootROM. The first stage, called bootstrap, does basic initialization of the switch processor and loads one of two second-stage bootloaders (called primary and secondary).

If the switch does not boot properly, both the bootstrap and the bootloader allow you to access the boot options using the CLI.

If necessary, the bootloader can be updated after the switch has booted, using TFTP. You can upgrade the bootROM from a TFTP server on the network after the switch has booted and only when asked to do so by an Extreme Networks technical representative.

For information about loading an image to a TFTP server and verifying which VR connects to your TFTP server, see Installing a Core Image on page 1707.

To download to a single stacking node, you need to specify the slot parameter:

download bootrom

Note
User-created VRs are supported only on the platforms listed for this feature in the licensing guide.

For ExtremeSwitching switches (except 5720 switches, which use ONIE), you can upgrade the bootROM with the version packaged in the ExtremeXOS .xos image file using the from-exos option: This command uses the bootROM image from the booted ExtremeXOS image. After reboot, the new bootROM image is used.

install bootrom

When you download and install new version of ExtremeXOS image, the system automatically compares the currently installed bootROM image against the bootROM image contained in the new ExtremeXOS image. If the installed version is older version, the system automatically upgrades bootROM to the version contained in new ExtremeXOS image. To view which version bootROM is on each partition, use the show version command.

If the bootROM image is invalid, this command shows 0.0.0.0. If the primary bootROM is invalid, the bootROM upgrade will be canceled. To fix this, re-install the bootROM on the primary partition.

Upgrading the Firmware

Firmware images are bundled with ExtremeXOS and Switch Engine software images. The bundled firmware images include system FPGA and PLD and microcontroller images. When you download and install a new version of ExtremeXOS, ExtremeXOS automatically compares the existing firmware image against the firmware images bundled with the ExtremeXOS image, and automatically upgrades the firmware if there is a new version. Upgraded firmware will become active after the next system reboot. If the firmware upgrade fails to become active after reboot, an EMS message will indicate the firmware version is not up to date and firmware can be installed by following the procedure below using the install firmware command.

Models with upgradeable firmware include: 4120, 4220, 5120, 5320, 5420, and 5520.

After a firmware image upgrade, messages are sent to the log.

To view the current firmware version, use the `show version detail` command:

```
# show version detail
Switch      : 800906-00-02 1907F-10042 Rev 02 IMG: 30.2.0.34
              FPGA: 1.1.34 CPLD: 1.1.13

Image   : ExtremeXOS version 30.2.0.34 by release-manager
          on Wed Mar 27 20:36:16 EDT 2019
Diagnostics : 1.1.6
Certified Version : EXOS Linux  4.14.107, FIPS fips-ecp-2.0.16
```

```
Build Tools Version : exos-x32-sdk-2.5.1.1.3
```

1. To install the firmware bundled with the ExtremeXOS image, use the `install firmware {`**`force`**`} {`**`slot`** *`slot-number`*`}` command.

   If the bundled firmware image is newer than the existing firmware image, the switch prompts you to confirm the upgrade.
2. At the prompt:
   - Type `y` to upgrade the firmware.
   - Type `n` to cancel the firmware upgrade for the specified hardware, and continue scanning for other hardware that needs to be upgraded.
3. Press **Enter** to cancel the upgrade.

   During the firmware upgrade, the switch also prompts you to save your configuration changes to the current, active configuration.
4. To save your configuration changes to the current, active configuration, type `y` or to discard your changes, type `n`.

   The new firmware overwrites the older version.

You must reboot the switch to load the newly installed firmware. However, you do not need to do this immediately after a firmware upgrade. Use the reboot command to reboot the switch and activate the new firmware. During the firmware upgrade, do not turn off or disrupt the power to the switch. During the firmware upgrade, do not cycle down or disrupt the power to the switch. If a power interruption occurs, the installed firmware may be corrupted. In this case, the switch uses a backup version, and you can run the upgrade again to install the latest version.

Displaying the BootROM and Firmware Versions

To display the bootROM (firmware) version on the switch and on all of the modules and PSU controllers, use the `show version` command.

The following is a sample for an 4120 Series switch:

```
4120-48MW-4Y-SwitchEngine.8 = show version
Switch          : 801187-00-AA FF022335G-00007 Rev AA BootROM: 4.6.0.3    IMG: 32.7.1.22

Image   : Extreme Networks Switch Engine version 32.7.1.22 by release-manager
          on Fri 12 Apr 2024 02:38:24 PM UTC

BootROM : Default 4.6.0.3  Alternate 4.6.0.3
Diagnostics : 0.0.1.6
Certified Version : EXOS Linux 5.10.177, Extreme Networks FIPS Object Module 2.0.16a
Build Tools Version : exos-arm64-sdk-3.1.19.1.0
```

The following is a sample for an 4220 Series switch:

```
4220-48P-4X-SwitchEngine.1 = show version
Switch          : 801146-00-03 PA062335G-00005 Rev 03 BootROM: 3.5.0.10   IMG: 32.7.0.527

Image   : Extreme Networks Switch Engine version 32.7.0.527 by release-manager
          on Mon 04 Mar 2024 04:42:03 PM UTC
BootROM : Default 3.5.0.10  Alternate 3.5.0.10
Diagnostics : 0.0.3.2
Certified Version : EXOS Linux 5.10.177, Extreme Networks FIPS Object Module 2.0.16a
Build Tools Version : exos-arm64-sdk-3.1.19.1.0
```

The following is a sample for an 5120 Series switch:

```
5120-44X-4Y-2C-SwitchEngine.8 = show version
Switch          : 801187-00-AA FF022335G-00007 Rev AA BootROM: 4.6.0.3    IMG: 33.1.1

Image   : Extreme Networks Switch Engine version 32.1.1 by release-manager
          on Fri 4 Apr 2025 02:38:24 PM UTC

BootROM : Default 4.6.0.3  Alternate 4.6.0.3
Diagnostics : 0.0.1.6
Certified Version : EXOS Linux 5.10.177, Extreme Networks FIPS Object Module 2.0.16a
Build Tools Version : exos-arm64-sdk-3.1.19.1.0
```

The following is a sample for a ExtremeSwitching 5520 series switch:

```
5520-48W-SwitchEngine.8 # show ver
Switch          : 800994-00-03 2008G-01290 Rev 03 BootROM: 2.2.1.8    IMG: 31.6.0.457
5520-VIM-4YE-1  : 800999-00-03 SB102011G-00319 Rev 03

Image   : Extreme Networks Switch Engine version 31.6.0.457 by release-manager
          on Mon 17 Jan 2022 07:44:41 PM UTC
BootROM : Default 2.2.1.8  Alternate 2.2.1.8
Diagnostics : 0.0.2.6
Certified Version : EXOS Linux 5.4.140, Extreme Networks FIPS Object Module 2.0.16a
Build Tools Version : exos-arm64-sdk-3.1.10.1.0
```

The following is a sample for a ExtremeSwitching 5420 series switch:

```
5420F-48T-4XE-SwitchEngine.1 # show version detail
Switch          : 801085-00-02 2039G-00608 Rev 02 BootROM: 2.3.2.1    IMG: 31.3.0.517
                  CPLD: 1.0.12 UC: 1.0.14

Image   : Switch Engine version 31.3.0.517 by release-manager
          on Thu Mar 11 10:43:08 EST 2021
BootROM : Default 2.3.2.1  Alternate 2.3.2.1
Diagnostics : 0.0.1.3
Certified Version : EXOS Linux 4.14.200, Extreme Networks FIPS Object Module 2.0.16a
Build Tools Version : exos-arm64-sdk-3.1.4.1.0
```

The following is a sample for a 5320 series switch:

```
5320-48P-SwitchEngine.8 # show ver
Switch          : 801085-00-02 2039G-00608 Rev 02 BootROM: 2.3.2.1    IMG: 31.3.0.517
                  CPLD: 1.0.12 UC: 1.0.14

Image   : Switch Engine version 31.3.0.517 by release-manager
          on Thu Mar 11 10:43:08 EST 2021
BootROM : Default 2.3.2.1  Alternate 2.3.2.1
Diagnostics : 0.0.1.3
Certified Version : EXOS Linux 4.14.200, Extreme Networks FIPS Object Module 2.0.16a
Build Tools Version : exos-arm64-sdk-3.1.4.1.0
```

The following is a sample for a ExtremeSwitching 5720 series switch:

```
5720-48MW-SwitchEngine.1 # show ver
Switch          : 801131-00-01 AR032207Q-40017 Rev 01 BootROM: N/A        IMG: 32.2.0.414

Image   : Extreme Networks Switch Engine version 32.2.0.414 by release-manager
          on Mon 12 Sep 2022 06:28:49 PM UTC
Diagnostics :
Certified Version : EXOS Linux 5.4.183, Extreme Networks FIPS Object Module 2.0.16i
Build Tools Version : exos-x86_64-sdk-3.1.14.1.0
```

The following is a sample for an ExtremeSwitching 7520 series switch:

```
7520-48Y-8C-SwitchEngine.1 # show ver
Switch          : 801131-00-01 AR032207Q-40017 Rev 01 BootROM: N/A        IMG: 32.4.0.414
```

```
Image   : Extreme Networks Switch Engine version 32.4.0.414 by release-manager
          on Mon 3 July 2023 06:28:49 PM UTC
Diagnostics :
Certified Version : EXOS Linux 5.4.183, Extreme Networks FIPS Object Module 2.0.16i
Build Tools Version : exos-x86_64-sdk-3.1.14.1.0
```

The following is a sample for an ExtremeSwitching 7720 series switch:

```
7720-32C-SwitchEngine.1 # show ver
Switch          : 801131-00-01 AR032207Q-40017 Rev 01 BootROM: N/A        IMG: 32.4.0.414

Image   : Extreme Networks Switch Engine version 32.4.0.414 by release-manager
          on Mon 3 July 2023 06:28:49 PM UTC
Diagnostics :
Certified Version : EXOS Linux 5.4.183, Extreme Networks FIPS Object Module 2.0.16i
Build Tools Version : exos-x86_64-sdk-3.1.14.1.0
```

Status Monitoring and Statistics

Viewing Port Statistics on page 1727
Viewing Port Errors on page 1728
Port Link-Flap Detection on page 1730
Using the Port Monitoring Display Keys on page 1732
Viewing VLAN Statistics on page 1733
Performing Switch Diagnostics on page 1734
Using the System Health Checker on page 1736
Using ELSM on page 1737
Viewing Fan Information on page 1749
Viewing the System Temperature on page 1749
Using the Event Management System/Logging on page 1750
Using the XML Notification Client on page 1767
Using sFlow on page 1770
Using RMON on page 1777
Monitoring CPU Utilization on page 1781
Non-Volatile Storage Health Monitoring on page 1783

Viewing statistics on a regular basis allows you to see how well your network is performing. If you keep simple daily records, you can see trends emerging and notice problems arising before they cause major network faults. In this way, statistics can help you get the best out of your network.

The status monitoring facility provides information about the switch.

This information may be useful for your technical support representative if you have a problem. ExtremeXOS software includes many command line interface (CLI) show commands that display information about different switch functions and facilities.

Note
For more information about show commands for a specific ExtremeXOS feature, see the appropriate chapter in this guide.

Viewing Port Statistics

ExtremeXOS software provides a facility for viewing port statistical information. The summary information lists values for the current counter for each port in the system. The switch automatically refreshes the display (this is the default behavior).

You can also display a snapshot of the real-time port statistics at the time you issue the command and view the output in a page-by-page mode. This setting is not saved; therefore, you must specify the no-refresh parameter each time you want a snapshot of the port statistics.

Values are displayed to nine digits of accuracy.

- View port statistics with the following command:

  `show ports {port_list | stack-ports stacking-port-list} statistics {no-refresh}`

  The switch collects the following port statistical information:

  - Link State—The current state of the link. Options are:
    - Active (A)—The link is present at this port.
    - Ready (R)—The port is ready to accept a link.
    - Loopback (L)—The port is configured for WANPHY loopback.
  - Transmitted Packet Count (TX Pkt Count)—The number of packets that have been successfully transmitted by the port.
  - Transmitted Byte Count (TX Byte Count)—The total number of data bytes successfully transmitted by the port.
  - Received Packet Count (RX Pkt Count)—The total number of good packets that have been received by the port.
  - Received Byte Count (RX Byte Count)—The total number of bytes that were received by the port, including bad or lost frames. This number includes bytes contained in the Frame Check Sequence (FCS), but excludes bytes in the preamble.
  - Received Broadcast (RX Bcast)—The total number of frames received by the port that are addressed to a broadcast address.
  - Received Multicast (RX Mcast)—The total number of frames received by the port that are addressed to a multicast address.
- View port statistics for SummitStack stacking ports with the following command:

  `show ports stack-ports {stacking-port-list} statistics {no-refresh}`

Viewing Port Errors

The switch keeps track of errors for each port and automatically refreshes the display (this is the default behavior). You can also display a snapshot of the port errors at the time you issue the command and view the output in a page-by-page mode. This setting is not saved; therefore, you must specify the *no-refresh* parameter each time you want a snapshot of the port errors.

- View port transmit errors with the following command:

  `show ports {port_list | stack-ports stacking-port-list} txerrors {no-refresh}`

  The switch collects the following port transmit error information:

  - Port Number—The number of the port.

  - Link State—The current state of the link. Options are:
    - Active (A)—The link is present at this port.
    - Ready (R)—The port is ready to accept a link.
    - Loopback (L)—The port is configured for WANPHY loopback.
  - Transmit Collisions (TX Coll)—The total number of collisions seen by the port, regardless of whether a device connected to the port participated in any of the collisions.
  - Transmit Late Collisions (TX Late Coll)—The total number of collisions that have occurred after the port's transmit window has expired.
  - Transmit Deferred Frames (TX Deferred)—The total number of frames that were transmitted by the port after the first transmission attempt was deferred by other network traffic.
  - Transmit Errored Frames (TX Errors)—The total number of frames that were not completely transmitted by the port because of network errors (such as late collisions or excessive collisions).
  - Transmit Lost Frames (TX Lost)—The total number of transmit frames that do not get completely transmitted because of buffer problems (FIFO underflow).
  - Transmit Parity Frames (TX Parity)—The bit summation has a parity mismatch.

- View port receive errors with the command:

  `show ports {`*`port_list`* `|` **`stack-ports`** *`stacking-port-list`*`}` **`rxerrors`** `{`**`no-refresh`**`}`

  The switch collects the following port receive error information:

  - Port Number
  - Link State—The current state of the link. Options are:
    - Active (A)—The link is present at this port.
    - Ready (R)—The port is ready to accept a link.
    - Loopback (L)—The port is in Loopback mode.
  - Receive Bad CRC Frames (RX CRC)—The total number of frames received by the port that were of the correct length but contained a bad FCS value.
  - Receive Oversize Frames (RX Over)—The total number of good frames received by the port greater than the supported maximum length of 1,522 bytes.
  - Receive Undersize Frames (RX Under)—The total number of frames received by the port that were less than 64 bytes long.
  - Receive Fragmented Frames (RX Frag)—The total number of frames received by the port that were of incorrect length and contained a bad FCS value.
  - Receive Jabber Frames (RX Jabber)—The total number of frames received by the port that were greater than the support maximum length and had a Cyclic Redundancy Check (CRC) error.
  - Receive Alignment Errors (RX Align)—The total number of frames received by the port with a CRC error and not containing an integral number of octets.
  - Receive Frames Lost (RX Lost)—The total number of frames received by the port that were lost because of buffer overflow in the switch.

- For SummitStack stacking ports, you can also view transmit and receive errors with the following commands:

  `show ports stack-ports` *`stacking-port-list`*`}` **`txerrors`** `{` **`no-refresh`**`}`

  `show ports stack-ports {`*`stacking-port-list`*`}` **`rxerrors`** `{` **`no-refresh`**`}`

  Information displayed is identical to the details displayed for non-stacking ports.

Port Link-Flap Detection

This feature detects excessive link flapping (link going up and down rapidly) on a physical port and can take the following actions:

- **Disables port**—After the port is disabled due to excessive link flapping, the port either stays down for the configured disable time value or can be re-enabled manually.
- **Reports issue to Syslog**—Adds an entry in the log for a link down event (see Link-Flap Detection Syslog Messages on page 1731).
- **Generates SNMP trap**—SNMP trap generated for a link down event (see SNMP Traps on page 1731).

You can configure the interval, threshold (maximum number of link down events), and disable time, and which of the preceding actions to take on a per port basis.

Link-Flap Detection Threshold Value Dependencies

Dependency on Hardware Link Scan Interval

The maximum value configurable for threshold depends on the link-flap interval value and the link scan interval. For example, if the link scan interval is 50 ms (default link-scan interval on most platforms), maximum link down/up events detectable by the hardware is 20 per second. Thus the maximum number of link-flaps detectable per second is 10. If the link-flap interval is 20 seconds, the maximum configurable threshold value is 20 * 10 = 200.

Thus, the maximum configurable value for threshold = link-flap interval (seconds) * maximum link-flaps detectable/second.

Dependency on Debounce Timer

The debounce timer is implemented at the chip level. Since the link-flap algorithm is implemented at a higher level in the *VLAN (Virtual LAN)* manager application, any link-flaps within the debounce timer are not reported to HAL nor the VLAN manager. For example, if the debounce timer is configured to be 400 ms, then any link-flaps within that interval are not reported to HAL and hence they are not counted.

Link-Flap Detection Syslog Messages

If you've configured link-flap detection to send log messages, the following messages are sent after excessive flapping:

- If the threshold is exceeded, and the log action is set, and the disable port action is not set, then a log message similar to the following appears:

```
03/31/2016 15:56:36.52 <Noti:vlan.msgs.PortLinkFlapActLogEvent> Port 1 link-flap
detection operational status is UP. Link has flapped more than 3 times in 60 seconds.
```

- If the log action and disable port action are set, and the threshold is exceeded, then a log message similar to the following appears:

```
03/31/2016 15:51:13.61 <Noti:vlan.msgs.PortLinkFlapActLogEvent> Port 2 link-flap
detection operational status is DOWN. Link has flapped more than 3 times in 60
seconds.
```

- This log message appears when all of the following conditions are met:
  - Disable port action is set on a port.
  - The number of link-flaps on the port exceeds the threshold.
  - You issue the command `enable log debug-mode`.
  - You add the event to the log filter (for example: "conf log filter "DefaultFilter" add events vlan.msgs.PortLinkFlapActDsblPort")

```
03/31/2016 15:51:13.61 <Summ:vlan.msgs.PortLinkFlapActDsblPort> Port 2 is disabled for
excessive link flapping. Link has flapped more than 3 times in 60 seconds.
```

SNMP Traps

If configured, after excessive link flapping an *SNMP (Simple Network Management Protocol)* trap notification is generated for link up or down events. The trap notification “etsysLinkFlapViolation” includes the following information regarding the port in violation of threshold:

- Interface index
- Interface name
- Operation status (up or down)

Port Link-Flap Statistics

Statistics	Description
Current Lin- Flaps	Number of link-flaps in the current interval before the threshold is exceeded.
Total Link-Flaps	Total number of link-flaps accumulated for this port since the feature was turned on.
Time Elapsed	Time in seconds that has passed in the current interval.
Threshold Exceeded	Number of times a given port has exceeded the threshold value for the maximum number of tolerated link-flaps per interval.

Configuring Port Link-Flap Detection

To turn on or off port link-flap detection, use the following command:

```
configure ports [port_list | all] link-flap-detection [on | off]
```

To set interval, threshold (maximum number of link down events), and disable time values for link-flap detection, use the following command:

```
configure ports [port_list | all] link-flap-detection [{interval
[interval | indefinitely]} {threshold threshold} {disable-time
[disable_time | until-cleared]}]
```

To reset interval, threshold (maximum number of link down events), and disable time to defaults for link-flap detection, use the following command.

```
unconfigure ports [port_list | all] link-flap-detection {interval}
{threshold} {disable-time}
```

To add or delete actions (disabling ports, logging events, generating *SNMP* traps) to be taken when excessive link-flapping is detected, use the following command:

```
configure ports [port_list | all] link-flap-detection action [add |
delete] [{{disable-port} {log} {trap}} | all-actions]
```

To manually enable ports that have been disabled due to excessive link flapping, use the following command:

```
clear ports [port_list | all] link-flap-detection status
```

To clears the counters related to port link flapping, use the following command.

```
clear ports [port_list | all] link-flap-detection counters
```

To display the link-flap detection status for ports, use the following command:

```
show ports [port_list | all] link-flap-detection [disabled-ports |
configuration | counters {refresh | no-refresh}]
```

Using the Port Monitoring Display Keys

Table 151 on page 1732 describes the keys used to control the displays that appear if you use any of the show ports commands without specifying the **no-refresh** parameter (this is the default behavior).

Table 151: Port Monitoring Display Keys with Auto-Refresh Enabled

Key(s)	Description
[U]	Displays the previous page of ports.
[D]	Displays the next page of ports.
[Esc]	Exits from the screen.

Table 151: Port Monitoring Display Keys with Auto-Refresh Enabled (continued)

Key(s)	Description
[0]	Clears all counters.
[Space]	Cycles through the following screens: • Packets per second • Bytes per second • Percentage of bandwidth **Note:** Available only using the `show ports utilization` command.

Table 152 describes the keys used to control the displays that appear if you use any of the show ports commands and specify the **no-refresh** parameter.

Table 152: Port Monitoring Display Keys with Auto-Refresh Disabled

Key	Description
[Q]	Exits from the screen.
[Space]	Displays the next page of ports.

Viewing VLAN Statistics

ExtremeXOS software provides the facility for configurable *VLAN* statistics gathering and display of packet and byte counters on the port level and on the VLAN level.

Configuring VLAN Statistics

- Configure the switch to start counting *VLAN* statistics with the commands:

  `clear counters`

  `configure ports [port_list|all] monitor vlan vlan_name {rx-only | tx-only}`

- View VLAN statistics at the port level with the command:

  `show ports {port_list} vlan statistics {no-refresh}`

  The switch collects and displays the following statistics:

  - Port—The designated port.
  - VLAN—The associated VLANs.
  - Rx Frames Count—The total number of frames successfully received by the port on the designated VLAN.
  - Rx Byte Count—The total number of bytes that were received by the port on the designated VLAN.
  - Tx Frame Count—The total number of frames that were transmitted by the port on the designated VLAN.
  - Tx Byte Count—The total number of bytes that were transmitted by the port on the designated VLAN.

Frame and byte counters are also displayed through *SNMP*.

- View VLAN statistics at the VLAN level with the command:

  `show ports {port_list} vlan statistics {no-refresh}`

  The switch collects and displays the following statistics:

  - VLAN—The designated VLAN.
  - Rx Total Frames —The total number of frames successfully received by the port.
  - Rx Byte Count—The total number of bytes that were received by the port.
  - Tx Total Frames—The total number of frames that were transmitted by the port.
  - Tx Byte Count—The total number of bytes that were transmitted by the port.
- Stop counting VLAN statistics.

  `unconfigure ports [port_list |all] monitor vlan vlan_name`

Note
While using VLAN statistics, traffic also matching egress *ACL (Access Control List)* counters will cause the VLAN statistics Tx counter to not increment.

Guidelines and Limitations

The following describes guidelines for this feature.

- Support for VMAN statistics are provided in the same manner as *VLAN* statistics. CLI commands use the same syntax as used with monitoring VLANs including the use of the vlan keyword in CLI commands. *SNMP* access uses the same MIB objects as used for VLAN statistics.

  Statistics for VLANs encapsulated within VMANs are not supported.
- ExtremeSwitching series switches provide support only for byte counters.
- All of the counters supported are 64 bit counters. No indication of counter rollover is supported.
- Packets originating from the switch's CPU or forwarded by the CPU may not be reflected in transmit statistics.

Performing Switch Diagnostics

The switch provides a facility for running normal or extended diagnostics.

In simple terms, a normal routine performs a simple ASIC and packet loopback test on all ports, and an extended routine performs extensive ASIC, ASIC-memory, and packet loopback tests. By running and viewing the results from diagnostic tests, you can detect and troubleshoot any hardware issues.

On ExtremeSwitching series switches, you run the diagnostic routine on the switch or on the stacking ports. Running the switch or stacking port diagnostic routine affects system operation; the switch is unavailable during the diagnostic test.

Note
Before running diagnostics, you must power on the External Power Supply (EPS) when it is connected to the switch.

When you run diagnostics, the switch verifies that the:

- Registers can be written to and read from correctly.
- Memory addresses are accessed correctly.
- Application-Specific Integrated Circuit (ASICs) and Central Processing Unit (CPUs) operate as required.
- External ports can send and receive packets.
- Sensors, hardware controllers, and LEDs are working correctly.

When you run diagnostics on the SummitStack stacking ports, the switch completes a hardware test to ensure that the stacking ports are operational.

Note
No ExtremeSwitching Universal switches include the diagnostic image in the main ExtremeXOS core software image (.xos). Instead, they have a separate diagnostic image that is upgraded using an XMOD file. For information about how to install an XMOD file, see Installing a Modular Software Package on page 1711.

Running Diagnostics

If you run the diagnostic routine, the switch reboots and then performs the diagnostic test.

During the test, traffic to and from the ports on the switch is temporarily unavailable. When the diagnostic test is complete, the switch reboots and becomes operational again.

To run the diagnostic routine on the stack ports, you need a dedicated stacking cable that connects stack port 1 to stack port 2, which are located at the rear of the switch. The stacking cable is available from Extreme Networks. The switch performs a hardware test to confirm that the stack ports are operational; traffic to and from the ports on the switch is temporarily unavailable. This Bit Error Rate Test (BERT) provides an analysis of the number of bits transmitted in error.

After the switch runs the diagnostic routine, test results saved to the switch's EEPROM and messages are logged to the syslog.

- Run diagnostics with the following command:

  `run diagnostics [extended | normal | stack-port] {slot [slot | A | B]}`

  Where the following is true:

extended—Reboots the switch and performs extensive ASIC, ASIC-memory, and packet loopback tests. Extended diagnostic tests take a maximum of five minutes. The CPU is not tested.

- normal—Reboots the switch and performs a simple ASIC and packet loopback test on all ports.
- stack-port—Performs a BERT on the stacking ports and reboots the switch.

LED Behavior During Diagnostic Testing

Table 153 describes the LED behavior during a diagnostic test.

Table 153: ExtremeSwitching Switch LED Behavior

LED	Color	Indicates
MGMT	Green blinking	Normal operation is occurring.
	Amber blinking	Diagnostic test in progress.

While diagnostic tests are running, the MGMT LED blinks amber. If a diagnostic test fails, the MGMT LED continues to blink amber. During normal operation, the MGMT LED blinks green.

Displaying Diagnostic Test Results

- Display the status of the last diagnostic test run on the switch.

  `show diagnostics {slot [slot_number]} {boot-time}`

Example

The following example displays the results of boot-time diagnostics on a specific stack slot:

```
# show diagnostics slot 2 boot-time

Slot-2: 5520-48SE-ACDC
Time:   Thu May 23 19:04:11 2024

Component             Result  Details
====================  ======  ================================================
CPU                   PASS
Memory                PASS
Storage               PASS
Power                 PASS    PSU-1: Powered On, PSU-2: Empty
Mgmt interface        PASS    Mgmt interface present, Link Up
```

Using the System Health Checker

The system health checker is a useful tool to monitor the overall health of your system.

The software performs a proactive, preventive search for problems by polling and reporting the health of system components, including power supplies, power supply controllers, and fans. By isolating faults to a specific module, backplane connection,

control plane, or component, the system health checker notifies you of a possible hardware fault.

The system health checker polls and reads the switch fabric and CPU registers.

Polling is always enabled on the system and occurs in the background every 10 seconds; the polling value is not a user-configured parameter.

System health check errors are reported to the Syslog. If you see an error, contact Extreme Networks Technical Support (see Help and Support). There are no health checking tests related to the stacking links in a SummitStack.

To display the system health check setting, including polling and how ExtremeXOS software handles faults on the switch, use the command: `show switch`

The system health check setting, displayed as SysHealth check, shows the polling setting and how ExtremeXOS handles faults. The polling setting appears as Enabled, and the fault handling setting appears in parenthesis next to the polling setting.

Using ELSM

Extreme Link Status Monitoring (ELSM) is an Extreme Networks proprietary protocol that monitors network health by detecting CPU and remote link failures.

ELSM is available only on Extreme Networks devices and operates on a point-to-point basis. You can configure ELSM on the ports that connect to other network devices and on both sides of the peer connection.

About ELSM

ELSM monitors network health by exchanging various hello messages between two ELSM peers.

ELSM uses an open-ended protocol, which means that an ELSM-enabled port expects to send and receive hello messages from its peer. The Layer 2 connection between ports determines the peer connection. Peers can be either directly connected or separated by one or more hubs. If there is a direct connection between peers, they are considered neighbors.

If ELSM detects a failure, the ELSM-enabled port responds by blocking traffic on that port. For example, if a peer stops receiving messages from its peer, ELSM brings down that connection by blocking all incoming and outgoing data traffic on the port and notifying applications that the link is down.

In some situations, a software or hardware fault may prevent the CPU from transmitting or receiving packets, thereby leading to the sudden failure of the CPU. If the CPU is unable to process or send packets, ELSM isolates the connections to the faulty switch from the rest of the network. If the switch fabric sends packets during a CPU failure, the switch may appear healthy when it is not. For example, if hardware forwarding is active and software forwarding experiences a failure, traffic forwarding may continue. Such failures can trigger control protocols such as *ESRP (Extreme*

Standby Router Protocol) or *EAPS (Extreme Automatic Protection Switching)* to select different devices to resume forwarding. This recovery action, combined with the CPU failure, can lead to loops in a Layer 2 network.

ELSM Hello Messages

ELSM uses two types of hello messages to communicate the health of the network to other ELSM ports. The following describes the hello messages:

- Hello+ — The ELSM-enabled port receives a hello message from its peer and no problem is detected.
- Hello- — The ELSM-enabled port has not received a hello message from its peer.

In addition to the ELSM port states described in the next section, ELSM has hello transmit states. The hello transmit states display the current state of transmitted ELSM hello messages and can be one of the following:

- HelloRx(+)—The ELSM-enabled port is up and receives Hello+ messages from its peer. The port remains in the HelloRx+ state and restarts the HelloRx timer each time it receives a Hello+ message. If the HelloRx timer expires, the hello transmit state enters HelloRx(-). The HelloRx timer is 6 * hello timer, which by default is 6 seconds.
- HelloRx(-)—The ELSM-enabled port either transitions from the initial ELSM state or is up but has not received hello messages because there is a problem with the link or the peer is missing.

For information about displaying ELSM hello messages and hello transmit states, see Displaying ELSM Information on page 1745.

ELSM Port States

Each ELSM-enabled port exists in one of the following states:

- Up—Indicates a healthy remote system and this port is receiving Hello+ messages from its peer.

  If an ELSM-enabled port enters the Up state, the up timer begins. Each time the port receives a Hello+ message from its peer, the up timer restarts and the port remains in the Up state. The up timer is 6 * hello timer, which by default is 6 seconds.
- Down—Indicates that the port is down, blocked, or has not received Hello+ messages from its peer.

  If an ELSM-enabled port does not receive a hello message from its peer before the up timer expires, the port transitions to the Down state. When ELSM is down, data packets are neither forwarded nor transmitted out of that port.
- Down-Wait—Indicates a transitional state.

  If the port enters the Down state and later receives a Hello+ message from its peer, the port enters the Down-Wait state. If the number of Hello+ messages received is greater than or equal to the hold threshold (by default, two messages), the port

transitions to the Up state. If the number of Hello+ messages received is less than the hold threshold, the port enters the Down state.

- Down-Stuck—Indicates that the port is down and requires user intervention.

  If the port repeatedly flaps between the Up and Down states, the port enters the Down-Stuck state. Depending on your configuration, there are two ways for a port to transition out of this state:

  By default, automatic restart is enabled, and the port automatically transitions out of this state. For more information, see the `enable elsm ports port_list` **`auto-restart`** command.

  If you disabled automatic restart, and the port enters the Down-Stuck state, you can clear the stuck state and enter the Down state by using one of the following commands:

  - `clear elsm ports port_list` **`auto-restart`**
  - `enable elsm ports port_list` **`auto-restart`**

Note

If you reboot the peer switch, its ELSM-enabled peer port may enter the Down-Stuck state. If this occurs, clear the stuck state using one of the following commands:

- `clear elsm ports port_list` **`auto-restart`**
- `enable elsm ports port_list` **`auto-restart`**

For information about displaying ELSM port states, see Displaying ELSM Information on page 1745.

Link States

The state of the link between ELSM-enabled (peer) ports is known as the *link state*. The link state can be one of the following:

- Ready—Indicates that the port is enabled but there is no physical link
- Active—Indicates that the port is enabled and the physical link is up

- View the state of the link between the peer ports using the commands:

  `show elsm ports` **`all`** `| port_list`

  `show ports {port_list}` **`information`** `{`**`detail`**`}`

  If you use the `show elsm ports` **`all`** `| port_list` command, the Link State row displays link state information.

  If you use the `show ports {port_list}` information command, the Link State column displays link state information.

  If you use the `show ports {port_list}` information command and specify the detail option, the ELSM Link State row displays ELSM link state information. For more information, see ELSM Link States on page 1740.

For more information about these show commands, see Displaying ELSM Information on page 1745.

ELSM Link States

The state of the ELSM logical link is known as the ELSM link state. The ELSM link state can be one of the following:

- ELSM is enabled and the ELSM peer ports are up and communicating
- ELSM is enabled but the ELSM peer ports are not up or communicating
- ELSM is disabled

- View the current state of ELSM on the switch using the commands:
  - `show elsm`

  `show elsm ports` **`all`** | *`port_list`*

  `show ports {`*`port_list`*`}` **`information`** `{`**`detail`**`}`

  If you use the `show elsm` commands, the following terms display the ELSM link state:

  - Up—Indicates that ELSM is enabled and the ELSM peer ports are up and communicating; the ELSM link state is up. In the up state, the ELSM-enabled port sends and receives hello messages from its peer.
  - Down—Indicates that ELSM is enabled, but the ELSM peers are not communicating; the ELSM link state is down. In the down state, ELSM transitions the peer port on this device to the down state. ELSM blocks all incoming and outgoing switching traffic and all control traffic except ELSM PDUs.

  If ELSM is disabled, the switch does not display any information.

  If you use the `show ports {`*`port_list`*`}` **`information`** `{`**`detail`**`}` command, the following columns display the current state of ELSM on the switch:

  - Flags
    - L—Indicates that ELSM is enabled on the switch
    - - —Indicates that ELSM is disabled on the switch
  - ELSM
    - up—Indicates that ELSM is enabled and the ELSM peer ports are up and communicating; the ELSM link state is up. In the up state, the ELSM-enabled port sends and receives hello messages from its peer.
    - dn—Indicates that ELSM is enabled, but the ELSM peers are not communicating; the ELSM link state is down. In the down state, ELSM transitions the peer port on this device to the down state. ELSM blocks all incoming and outgoing switching traffic and all control traffic except ELSM PDUs.
    - - —Indicates that ELSM is disabled on the switch.

If you specify the optional detail parameter, the following ELSM output is called out in written explanations versus displayed in a tabular format:

- ELSM Link State (displayed only if ELSM is enabled on the switch)
  - Up—Indicates that ELSM is enabled and the ELSM peer ports are up and communicating; the ELSM link state is up. In the up state, the ELSM-enabled port sends and receives hello messages from its peer.
  - Down—Indicates that ELSM is enabled, but the ELSM peers are not communicating; the ELSM link state is down. In the down state, ELSM transitions the peer port on this device to the down state. ELSM blocks all incoming and outgoing switching traffic and all control traffic except ELSM PDUs.
- ELSM
  - Enabled—Indicates that ELSM is enabled on the switch
  - Disabled—Indicates that ELSM is disabled on the switch

For more information about these show commands, see Displaying ELSM Information on page 1745.

ELSM Timers

To determine whether there is a CPU or link failure, ELSM requires timer expiration between the ELSM peers. Depending on the health of the network, the port enters different states when the timers expire. For more information about the ELSM port states, see ELSM Port States on page 1738.

Table 154 describes the ELSM timers. Only the hello timer is user-configurable; all other timers are derived from the hello timer. This means that when you modify the hello timer, you also modify the values for down, up, and HelloRx timers.

Table 154: ELSM Timers

Timer	Description
Hello	The ELSM hello timer is the only user-configurable timer and specifies the time in seconds between consecutive hello messages. The default value is 1 second, and the range is 100 milliseconds to 255 seconds.
Down	The ELSM down timer specifies the time it takes the ELSM port to cycle through the following states: • Down—Indicates that the port is down, blocked, or has not received Hello+ messages from its peer. • Down-Wait—Indicates a transitional state.Up—Indicates a healthy remote system and this port is receiving Hello+ messages from its peer. By default, the down timer is (2 + hold threshold) * hello timer, which is 4 seconds. If the hold threshold is set to 2 and the hello timer is set to 1 second, it takes 4 seconds for the ELSM port receiving messages to cycle through the states. After the down timer expires, the port checks the number of Hello+ messages against the hold threshold. If the number of Hello+ messages received is greater than or equal to the configured hold threshold, the ELSM receive port moves from the Down-Wait state to the Up state. If the number of Hello+ messages received is less than the configured hold threshold, the ELSM receive port moves from the Down-Wait state back to the Down state and begins the process again.
Up	The ELSM up timer begins when the ELSM-enabled port enters the UP state. Each time the port receives a Hello+ message, the timer restarts. Up timer is the UpTimer threshold * hello timer. It is configurable and the range is 3 to 60 seconds. By default, the UpTimer threshold is 6. Therefore, the default up timer is 6 seconds (6*1).
HelloRx	The ELSM HelloRx timer specifies the time in which a hello message is expected. If the port does not receive a hello message from its peer, there is the possibility of a CPU or link failure. By default the HelloRx timer 6 * hello timer, which is 6 seconds.

Configuring ELSM on a Switch

This section describes the commands used to configure ELSM on the switch.

Note
ELSM and mirroring are mutually exclusive. You can enable either ELSM, or mirroring, but not both.

Note
ELSM is not to be configured on ports at either end of an MLAG.

Enabling ELSM

ELSM works between two connected ports (peers), and each ELSM instance is based on a single port. The Layer 2 connection between the ports determines the peer. You can have a direct connection between the peers or hubs that separate peer ports. In the first instance, the peers are also considered neighbors. In the second instance, the peer is not considered a neighbor.

- Enable ELSM on one or more ports with the command:

  ```
  enable elsm ports port_list
  ```

When you enable ELSM on a port, ELSM immediately blocks the port and it enters the Down state. When the port detects an ELSM-enabled peer, the peer ports exchange ELSM hello messages. At this point, the ports enter the transitional Down-Wait state. If the port receives Hello+ messages from its peer and does not detect a problem, the peers enter the Up state. If a peer detects a problem or there is no peer port configured, the port enters the Down state.

For more information about the types of ELSM hello messages, see ELSM Hello Messages on page 1738. For information about configuring the ELSM hello timer, see the next section.

Configuring the ELSM Hello Timer

The ELSM hello timer is the only user-configurable timer and specifies the time in seconds between consecutive hello messages. The default value is 1 second, and the range is 1 to 128 seconds. Although other timers rely on the hold timer for their values, you do not explicitly configure the down, up, or HelloRx timers. If you modify the hello timer on one port, we recommend that you use the same hello timer value on its peer port.

A high hello timer value can increase the time it takes for the ELSM-enabled port to enter the Up state. The down timer is (2 + hold threshold) * hello timer. Assuming the default value of 2 for the hold threshold, configuring a hello timer of 128 seconds creates a down timer of (2 + 2) 128, or 512 seconds. In this scenario it would take 512 seconds for the port to transition from the Down to the Up state.

- Configure the ELSM hello timer with the command:

  ```
  configure elsm ports port_list hellotime hello_time
  ```

Configuring the ELSM Hold Threshold

The ELSM hold threshold determines the number of Hello+ messages the ELSM peer port must receive to transition from the Down-Wait state to the Up state. For example, a threshold of 1 means the ELSM port must receive at least one Hello+ message to transition from the Down-Wait state to the Up state. The default is two messages, and the range is one to three messages.

After the down timer expires, the port checks the number of Hello+ messages against the hold threshold. If the number of Hello+ messages received is greater than or equal to the configured hold threshold, the ELSM receive port moves from the Down-Wait state to the Up state.

If the number of Hello+ messages received is less than the configured hold threshold, the ELSM receive port moves from the Down-Wait state back to the Down state and begins the process again.

If you modify the hold threshold on one port, we recommend that you use the same hold threshold value on its peer port.

You must configure the hold threshold on a per-port basis, not on a per-switch basis.

- Configure the ELSM hold threshold with the command:

  `configure elsm ports` *`port_list`* **`hold-threshold`** *`hold_threshold`*

Configure Automatic Restart

You must explicitly configure automatic restart on each ELSM-enabled port; this is not a global configuration. By default, ELSM automatic restart is enabled. If an ELSM-enabled port goes down, ELSM bypasses the Down-Stuck state and automatically transitions the down port to the Down state, regardless of the number of times the port goes up and down. For information about the port states, see ELSM Port States on page 1738.

If you disable ELSM automatic restart, the ELSM-enabled port can transition between the following states multiple times: Up, Down, and Down-Wait. When the number of state transitions is greater than or equal to the sticky threshold, the port enters the Down-Stuck state. The ELSM sticky threshold specifies the number of times a port can transition between the Up and Down states. The sticky threshold is not user-configurable and has a default value of 1. That means a port can transition only one time from the Up state to the Down state. If the port attempts a subsequent transition from the Up state to the Down state, the port enters the Down-Stuck state.

The following user events clear or re-set the sticky threshold:

- Enabling automatic restart on the port using the following command:

  `enable elsm ports` *`port_list`* **`auto-restart`**
- Clearing the port that is in the Down-Stuck state using the following command:

  `clear elsm ports` *`port_list`* **`auto-restart`**
- Disabling and re-enabling the port using the following commands:
  - `disable ports [`*`port_list`*`|`**`all`**`]`
  - `enable ports [`*`port_list`*`|`**`all`**`]`
- Disabling ELSM on the port using the following command:

  `disable elsm ports` *`port_list`*

We recommend that you use the same automatic restart configuration on each peer port.

- If the port enters the Down-Stuck state, you can clear the stuck state and enter the Down state by using one of the following commands:

  `clear elsm ports` *`port_list`* **`auto-restart`**

  `enable elsm ports` *`port_list`* **`auto-restart`**

If you use the `enable elsm ports port_list` **`auto-restart`** command, automatic restart is always enabled; you do not have to use the `clear elsm ports port_list` **`auto-restart`** command to clear the stuck state.

- Disable automatic restart with the command:

  `disable elsm ports port_list` **`auto-restart`**

- Re-enable automatic restart with the command:

  `enable elsm ports port_list` **`auto-restart`**

  We recommend that you use the same automatic restart configuration on each peer port.

Disabling ELSM

- Disable ELSM on one or more ports with the command:

  `disable elsm ports port_list`

When you disable ELSM on the specified ports, the ports no longer send ELSM hello messages to their peers and no longer maintain ELSM states.

Displaying ELSM Information

- Display summary information for all of the ELSM-enabled ports on the switch with the command:

  `show elsm`

  This command displays in a tabular format the operational state of ELSM on the configured ports.

  If ports are configured for ELSM (ELSM is enabled), the switch displays the following information:

  - Port—The port number of the ELSM-enabled port.
  - ELSM State—The current state of ELSM on the port. For information about the port states, see ELSM Port States on page 1738.
  - Hello time—The current value of the hello timer, which by default is 1 second. For information about configuring the hello timer, see Configuring the ELSM Hello Timer on page 1743.

  If no ports are configured for ELSM (ELSM is disabled), the switch does not display any information.

- Display detailed information for one or more ELSM-enabled ports on the switch.

  `show elsm ports` **`all`** | *`port_list`*

  In addition to the port, ELSM state, and hello timer information, this command displays in a tabular format the following:

  - Link State—The state of the link between ELSM-enabled ports. For information about the link states, see Link States on page 1739.
  - ELSM Link State—The current state of the ELSM logical link on the switch. For more information, see ELSM Link States on page 1740.

- Hello Transmit State—The current state of ELSM hello messages being transmitted.
- Hold Threshold—The number of Hello+ messages required by the ELSM-enabled port to transition from the Down-Wait state to the Up state within the hold threshold.
- UpTimer Threshold—The number of hello times that span without receiving Hello+ packets before a port changes its ELSM state from Up to Down.
- Auto Restart—The current state of ELSM automatic restart on the port.
- Sticky Threshold—The number of times a port can transition between the Up and Down states. The sticky threshold is not user-configurable and has a default value of 1.
- Sticky Threshold Counter—The number of times the port transitions from the Up state to the Down state.
- Down Timeout—The actual waiting time (msecs or secs) before a port changes its ELSM state from Down to Up after receiving the first Hello+ packet. It is equal to [Hello Time * (Hold Threshold+2)].
- Up Timeout—The actual waiting time (msecs or secs) before a port changes its ELSM state from Up to Down after receiving the last Hello+ packets. It is equal to [Hello Time * UpTimer Threshold].

The remaining output displays counter information.

- Use the counter information to determine the health of the ELSM peers and how often ELSM has gone up or down.

  The counters are cumulative.

  - Rx Hello+—The number of Hello+ messages received by the port.
  - Rx Hello- —The number of Hello- messages received by the port.
  - Tx Hello+—The number of Hello+ messages sent by the port.
  - Tx Hello- —The number of Hello- messages sent by the port.
  - ELSM Up/Down Count—The number of times ELSM has been up or down.
- Display summary port information in a tabular format for one or more ELSM-enabled ports with the command:

  `show ports {port_list} information {detail}`

  This command displays the following ELSM information:

  - Flags
    - L—ELSM is enabled on the switch
    - - —ELSM is disabled on the switch
  - ELSM (For more information, see ELSM Link States on page 1740.)
    - up—ELSM is enabled and the ELSM link state is up.
    - dn—ELSM is enabled and the ELSM link state is down.
    - - —ELSM is disabled on the switch.
- Display summary port information called out in written explanations versus displayed in a tabular format, use the detail parameter.

  `show ports {port_list} information {detail}`

This command displays the following ELSM information:

- ELSM Link State (displayed only if ELSM is enabled on the switch). For more information, see ELSM Link States on page 1740.
  - Up—ELSM is enabled and the ELSM link state is up.
  - Down—ELSM is enabled and the ELSM link state is down.
- Link State—The state of the link between ELSM-enabled ports. For information about the link states, see Link States on page 1739.
- ELSM
  - Enabled—ELSM is enabled on the switch.
  - Disabled—ELSM is disabled on the switch.

Clearing ELSM Counters

Before clearing the ELSM counters, you should use the `show elsm` and `show elsm ports` commands to view the ELSM information.

- Clear only the ELSM-related counters gathered by the switch using the command:
  `clear elsm {`**`ports`** *`port_list`*`}` **`counters`**

  You can also use the `clear counters` command, which clears all of the counters on the device, including those associated with ELSM.

Using ELSM with Layer 2 Control Protocols

You can use ELSM with Layer 2 control protocols such as *STP (Spanning Tree Protocol)*, *ESRP*, *EAPS*, and so on to improve the recovery of Layer 2 loops in the network.

ELSM detects remote link failures if the established link is through a Layer 2 transport cloud and the ELSM endpoints traverse the Layer 2 cloud in a point-to-point fashion, as shown in Figure 200 on page 1747.

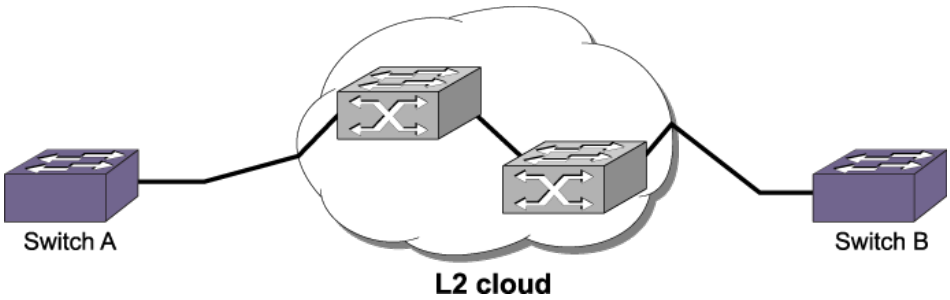


Figure 200: ELSM Through a Layer 2 Cloud

A sudden failure of the switch CPU may cause the hardware to continue forwarding traffic. For example, ESRP may select a second device for forwarding traffic thereby creating a Layer 2 loop in the network. If you configure ELSM and the CPU fails, ELSM closes the connection to the faulty device to prevent a loop.

ELSM Configuration Example

The following example configures ELSM on two ports connected directly to each other and assumes the following:

Note
In the following sample configurations, any lines marked (Default) represent default settings and do not need to be explicitly configured.

Switch A Configuration

- ELSM-enabled port—Port 1
- Hello timer—2 seconds
- Hello threshold—2 hello messages

```
enable elsm ports 1
configure elsm ports 1 hellotime 2
configure elsm ports 1 hold-threshold 2 (Default)
```

Switch B Configuration

- ELSM-enabled port—Slot 2, port 1
- Hello timer—2 seconds
- Hello threshold—2 hello messages

```
enable elsm ports 2:1
configure elsm ports 2:1 hellotime 2
configure elsm ports 2:1 hold-threshold 2 (Default)
```

After you enable ELSM on the ports, the peers exchange hello messages with each other as displayed in Figure 201.

Figure 201: Extreme Networks Switches with ELSM-Enabled Ports Exchanging Hello Messages

To configure ELSM with *LAG (Link Aggregation Group)*, enable ELSM on each port member of the LAG, as shown in the following example:

```
enable sharing 7 grouping 7-9, 41 algorith address-based L2
enable elsm port 7
enable elsm port 8
enable elsm port 9
enable elsm port 41

# sh elsm
Port ELSM State Hello Time
==== ========== ==========
7    Up         1 (secs)
8    Down       1 (secs)
9    Down       1 (secs)
41   Up         1 (secs)
```

Viewing Fan Information

You can view detailed information about the fans installed in your switch. Depending on your switch model, different information may be displayed.

- View detailed information about the health of the fans with the command: `show fans`

  The switch collects and displays the following fan information:

  - State—The current state of the fan. Options are:
    - Empty: There is no fan installed.
    - Failed: The fan failed.
    - Operational: The fan is installed and working normally.
  - NumFan—The number of fans in the fan tray.
  - Fan Name, displayed as Fan-1, Fan-2, and so on—Specifies the individual state for each fan in a fan tray and its current speed in revolutions per minute (rpm).

Viewing the System Temperature

Depending on your switch model, you can view the temperature in Celsius for the power controllers, power supplies, and fan trays installed in your switch. In addition, depending on the software version running on your switch, additional or different temperature information might be displayed, such as the inlet air temperature of a switch using an Extreme-supplied USB dongle temperature sensor for version 33.3.1 and later.

You can view the system temperature with the command: `show temperature`

System Temperature Output

SummitStack

The following output shows a sample display from a SummitStack.

```
Slot-3 Stack.1 # show temperature
Field Replaceable Units               Temp (C)   Status   Min  Normal  Max
-------------------------------------------------------------------------
Slot-1          :
Slot-2          : SummitX                34.50    Normal   -10    0-54  59
Slot-3          : SummitX                36.50    Normal   -10    0-66  67
Slot-4          :
Slot-5          :
Slot-6          :
Slot-7          :
Slot-8          :
Slot-3 Stack.2 #
```

ExtremeSwitching Switches Only

On ExtremeSwitching series switches, the output includes the current temperature, operating status of the switch, and inlet air temperature. The following shows a sample display from a switch:

```
# show temperature
Field Replaceable Units                                  Temp (C)   Status   Min
Normal   Max
-------------------------------------------------------  ---------  -------  ---
-------  ---
Switch                : 5520-24W                           68.38    Normal     0
10-100  110
Switch                : Inlet Air                          21.50
```

The switch monitors its temperature and generates a warning if the temperature exceeds the normal operating range. If the temperature exceeds the maximum limit, the `show switch` output indicates the switch in an OPERATIONAL (Overheat) mode, and the `show temperature` output indicates an error state due to overheat.

For the inlet air temperature readings, for switch models that support two USB ports, the sensor works in either port, but only one sensor will work at a time if two are plugged in. If two are installed and one is removed, the second one must be re-seated for the kernel/operating system to detect and use it.

Using the Event Management System/Logging

We use the general term *event* for any type of occurrence on a switch that could generate a log message or require an action.

For example, a link going down, a user logging in, a command entered on the command line, or the software executing a debugging statement, are all events that might generate a log message. The system for saving, displaying, and filtering events is called the EMS. With EMS, you have many options about which events generate log messages, where the messages are sent, and how they are displayed.

Using EMS you can:

- Send event messages to a number of logging targets (for example, syslog host and NVRAM).
- Filter events per target, by:
  - Component, subcomponent, or specific condition (for example, *BGP (Border Gateway Protocol)* messages, *IGMP (Internet Group Management Protocol)* Snooping messages, or the IP.Forwarding.SlowPathDrop condition).
  - Match expression (for example, any messages containing the string “user 5”).
  - Matching parameters (for example, only messages with source IP addresses in the 10.1.2.0/24 subnet).
  - Severity level (for example, only messages of severity critical, error, or warning).
- Change the format of event messages (for example, display the date as “12-May-2005” or “2005-05-12”).
- Display log messages in real time and filter the messages that are displayed, both on the console and from Telnet sessions.

- Display stored log messages from the memory buffer or NVRAM.
- Upload event logs stored in memory buffer or NVRAM to a TFTP server.
- Display counts of event occurrences, even those not included in filter.
- Display debug information using a consistent configuration method.

EMS supports IPv6 as a parameter for filtering events.

ExtremeCloud IQ Agent logging defaults to in-memory, buffer-based logging instead of continuous on-disk file logging, reducing storage consumption. Buffer logs are included in `show tech-support` output for diagnostics. File-based logging is disabled by default and available on demand via debug CLI. After upgrading to version 33.7.1, file-based logging is off by default, but existing log files are preserved.

Sending Event Messages to Log Targets

You can specify seven types of targets to receive log messages:

- Console display
- Current session (Telnet or console display)
- Memory buffer (can contain 200 to 20,000 messages)
- NVRAM (messages remain after reboot)
- Primary node (for SummitStack)
- Backup node (for SummitStack)
- Syslog server, supports IPv6 address family as of ExtremeXOS 21.1

The first six targets exist by default; but before enabling any syslog host, you must add the host's information to the switch using the `configure syslog` command.

By default, the memory buffer and NVRAM targets are already enabled and receive messages.

- To start sending messages to the targets, use the following command:

  `enable log target [`**`console`** | **`memory-buffer`** | **`nvram`** | **`primary-node`**|**`backup-node`**| **`session`** | **`syslog`** [**`all`** | *`ipaddress`* **`udp-port`** {*`udp_port`*} | *`ipPort`* | *`ipaddress`* **`tls_port`** {*`tls_port`*}] {**`vr`** *`vr_name`*} {**`local0...local7`**}]]

  After you enable this feature, the target receives the messages for which it is configured. See Target Configuration on page 1752 for information on viewing the current configuration of a target. The memory buffer can contain only the configured number of messages, so the oldest message is lost when a new message arrives, when the buffer is full.
- To stop sending messages to the target, use the following command:

  `disable log target [`**`console`** | **`memory-buffer`** | **`nvram`** |**`primary-node`** | **`backup-node`** | **`session`** | **`syslog`** [**`all`** | *`ipaddress`* **`udp-port`** {*`udp_port`*} | *`ipPort`* | *`ipaddress`* **`tls_port`** {*`tls_port`*}] {**`vr`** *`vr_name`*} {**`local0 ... local7`**}]]

Refer to your UNIX documentation for more information about the syslog host facility.

Primary and Backup Systems—SummitStack Only

A system with primary and backup nodes (SummitStack) keeps the two systems synchronized by executing the same commands on both.

However, the full data between the EMS servers is not synchronized. The reason for this design decision is to make sure that the control channel is not overloaded when a high number of log messages are generated.

To capture events generated by the primary node onto the backup node, two additional targets are shown in the target commands—one called primary-node and one called backup-node. The first target is active only on the non-primary (backup) EMS server and is used to send matching events to the primary EMS server. The other target is active only on the primary EMS server and is used to send matching events to all other EMS servers.

If the condition for the backup target is met by a message generated on the primary node, the event is sent to the backup node. When the backup node receives the event, it detects if any of the local targets (NVRAM, memory, or console) are matched. If so that event gets processed. The session and Syslog targets are disabled on the backup node, as they are handled on the primary. If the condition for the primary target is met by a message generated on the backup, the event is sent to the primary node.

Note that the backup target is active only on the primary node, and the primary target is active only on the backup node.

Filtering Events Sent to Targets

Not all event messages are sent to every enabled target. Each target receives only the messages for which it is configured.

Target Configuration

To specify the messages to send to an enabled target, you can set a message severity level, a filter name, and a match expression. These items determine which messages are sent to the target. You can also configure the format of the messages in the targets. For example, the console display target is configured to get messages of severity info and greater, the NVRAM target gets messages of severity warning and greater, and the memory buffer target gets messages of severity debug-data and greater. All the targets are associated by default with a filter named DefaultFilter that passes all events at or above the default severity threshold. All the targets are also associated with a default match expression that matches any messages (the expression that matches any message is displayed as Match : (none) from the command line). And finally, each target has a format associated with it.

- To display the current log configuration of the targets, use the following command:

  ```
  show log configuration target {console | memory-buffer | nvram |
  primary-node | backup-node | session | syslog {ipaddress {udp-port
  {udp_port }}| ipPort |ipaddress tls-port {tls_port}} {vr vr_name}
  {[local0...local7]}}
  ```

- To configure a target, you use specific commands for severity, filters, and formats, use the following command:

  ```
  configure log target [console | memory-buffer | nvram |primary-node |backup-node | session | syslog [all | ipaddress {udp-port {udp_port}} | ipPort | ipaddress tls-port {tls_port} ] {vr vr_name} {local0...local7 }] {severity severity {only}}
  ```

  In addition, you can configure the source IP address for a syslog target. Configuring the source IP address allows the management station or syslog server to identify from which switch it received the log messages.
- To configure the source IP address for a syslog target, use the following command.

  ```
  configure log target syslog [all | ipaddress {udp-port {udp_port}} | ipPort | ipaddress tls-port {tls_port}] {vr vr_name} {local0...local7} from source-ip-address
  ```

  Note
  The address family (i.e IPv4 or IPv6) of the specified source IP address must be the same as the address family of the syslog server.

Severity

Messages are issued with one of the following severity levels: Critical, Error, Warning, Notice, Info, Debug-Summary, Debug-Verbose, or Debug-Data, which are described in Table 155.

When a message is sent to a syslog target, the severity is mapped to a corresponding syslog priority value (see RFC 3164).

The three severity levels for extended debugging—Debug-Summary, Debug-Verbose, and Debug-Data—require that log debug mode be enabled (which may cause a performance degradation).

See Displaying Debug Information on page 1765 for more information about debugging.

Table 155: Severity Levels Assigned by the Switch

Level	Description
Critical	A serious problem has been detected that is compromising the operation of the system; the system cannot function as expected unless the situation is remedied. The switch may need to be reset.
Error	A problem has been detected that is interfering with the normal operation of the system; the system is not functioning as expected.
Warning	An abnormal condition, not interfering with the normal operation of the system, has been detected that indicate that the system or the network in general may not be functioning as expected.
Notice	A normal but significant condition has been detected, which signals that the system is functioning as expected.

Table 155: Severity Levels Assigned by the Switch (continued)

Level	Description
Info (Informational)	A normal but potentially interesting condition has been detected, which signals that the system is functioning as expected; this level simply provides potentially detailed information or confirmation.
Debug-Summary	A condition has been detected that may interest a developer seeking the reason underlying some system behavior.
Debug-Verbose	A condition has been detected that may interest a developer analyzing some system behavior at a more verbose level than provided by the debug summary information.
Debug-Data	A condition has been detected that may interest a developer inspecting the data underlying some system behavior.

Configuring Severity Level

You can use more than one command to configure the severity level of the messages sent to a target.

- The most direct way to set the severity level of all the sent messages is to use the following command:

  ```
  configure log target [console | memory-buffer | nvram |primary-
  node |backup-node | session | syslog [all | ipaddress {udp-port
  {udp_port}} | ipPort | ipaddress tls-port {tls_port} ] {vr vr_name}
  {local0...local7 }] {severity severity {only}}
  ```

  When you specify a severity level, messages of that severity level and greater are sent to the target. If you want only those messages of the specified severity to be sent to the target, use the keyword only. For example, specifying severity warning will send warning, error, and critical messages to the target, but specifying severity warning only sends only warning messages.

- You can also use the following command to configure severity levels, which associate a filter with a target:

  ```
  configure log target [console | memory-buffer | | primary-node | |
  backup-node | nvram | session | syslog [all | ipaddress {udp-port
  {udp_port}} | ipPort | ipaddress tls-port {tls_port} ]{vr vr_name}
  {local0...local7}] filter filter-name {severity severity {only}}
  ```

When you specify a severity level as you associate a filter with a target, you further restrict the messages reaching that target. The filter may allow only certain categories of messages to pass. Only the messages that pass the filter and then pass the specified severity level reach the target.

Finally, you can specify the severity levels of messages that reach the target by associating a filter with a target. The filter can specify exactly which message it will pass. Constructing a filter is described in Filtering By Components and Conditions on page 1756.

Components and Conditions

The event conditions detected by ExtremeXOS are organized into components and subcomponents.

- To get a listing of the components and subcomponents in your release of ExtremeXOS, use the following command:

  `show log components {event component } {version}`

  For example, to get a list of the components and subcomponents in your system, use the following command: `show log components`

  The following is partial output from this command:

```
Severity
Component           Title                                          Threshold
------------------- ---------------------------------------------- -------------
...
...
STP                 Spanning-Tree Protocol (STP)                   Error
InBPDU              STP In BPDU subcomponent                       Warning
OutBPDU             STP Out BPDU subcomponent                      Warning
System              STP System subcomponent                        Error
...
...
```

  The display above lists the components, subcomponents, and the severity threshold assigned to each. In EMS, you use a period (.) to separate component, subcomponent, and condition names. For example, you can refer to the InBPDU subcomponent of the *STP* component as STP.InBPDU. On the CLI, you can abbreviate or **[Tab]** complete any of these.

  A component or subcomponent typically has several conditions associated with it.

- To see the conditions associated with a component, use the following command:

  `show log events [event condition | [all | event component] {severity severity {only}}] {details}`

  For example, to see the conditions associated with the STP.InBPDU subcomponent, use the following command: `show log events stp.inbpdu`

  The following is sample output from this command:

```
Comp    SubComp     Condition              Severity      Parameters
------- ----------- ---------------------- ------------- ----------
STP     InBPDU      Drop                   Error          2 total
STP     InBPDU      Dump                   Debug-Data     3 total
STP     InBPDU      Trace                  Debug-Verbose  2 total
STP     InBPDU      Ign                    Debug-Summary  2 total
STP     InBPDU      Mismatch               Warning        2 total
```

  The display above lists the five conditions contained in the STP.InBPDU component, the severity of the condition, and the number of parameters in the event message. In this example, the severities of the events in the STP.InBPDU subcomponent range from error to debug-summary.

When you use the **details** keyword, you see the message text associated with the conditions.

- For example, if you want to see the message text and the parameters for the event condition STP.InBPDU.Trace, use the following command: `show log events stp.inbpdu.trace details`

  The following is sample output from this command:

```
Comp    SubComp     Condition               Severity      Parameters
------- ----------- ----------------------- ------------- ----------
STP     InBPDU      Trace                   Debug-Verbose  2 total
0 - string
1 - string (printf)
Port=%0%: %1%
```

  The Comp heading shows the component name, the SubComp heading shows the subcomponent (if any), the Condition heading shows the event condition, the Severity heading shows the severity assigned to this condition, the Parameters heading shows the parameters for the condition, and the text string shows the message that the condition will generate. The parameters in the text string (for example, %0% and %1% above) will be replaced by the values of these parameters when the condition is encountered and displayed as the event message.

Filtering By Components and Conditions

You may want to send the messages that come from a specific component that makes up ExtremeXOS or to send the message generated by a specific condition. For example, you might want to send only those messages that come from the *STP* component, or send the message that occurs when the IP.Forwarding.SlowPathDrop condition occurs. Or you may want to exclude messages from a particular component or event. To do this, you construct a filter that passes only the items of interest, and you associate that filter with a target.

1. The first step is to create the filter using the `create log filter` command.

   You can create a filter from scratch, or copy another filter to use as a starting point. (It may be easiest to copy an existing filter and modify it.)
2. To create a filter, use the following command:

   `create log filter` *name* {**copy** *filter_name*}

   If you create a filter from scratch, that filter initially blocks all events until you add events (either the events from a component or a specific event condition) to pass. You might create a filter from scratch if you want to pass a small set of events and to block most events. If you want to exclude a small set of events, use the default filter that passes events at or above the default severity threshold (unless the filter has been modified), named DefaultFilter, that you can copy to use as a starting point for your filter.
3. After you create your filter, you configure filter items that include or exclude events from the filter.

   Included events are passed; excluded events are blocked.

4. To configure your filter, use the following command:

   ```
   configure log filter name [add | delete] {exclude} events [event-condition | [all | event-component] {severity severity {only}}]
   ```

   For example, if you create the filter myFilter from scratch, use the following command to include events:

   ```
   configure log filter myFilter add events stp
   ```

   All STP component events of at least the default threshold severity passes myFilter (for the STP component, the default severity threshold is error). You can further modify this filter by specifying additional conditions.

   For example, assume that myFilter is configured as before, and assume that you want to exclude the STP.CreatPortMsgFail event.

5. To add that condition, use the following command:

   ```
   configure log filter myFilter add exclude events stp.creatportmsgfail
   ```

6. You can also add events and subcomponents to the filter.

   For example, assume that myFilter is configured as before, and you want to include the STP.InBPDU subcomponent. To add that condition, use the following command:

   ```
   configure log filter myFilter add events stp.inbpdu
   ```

7. You can continue to modify this filter by adding more filter items.

   The filters process events by comparing the event with the most recently configured filter item first. If the event matches this filter item, the incident is either included or excluded, depending on whether the exclude keyword was used. If necessary, subsequent filter items on the list are compared. If the list of filter items is exhausted with no match, the event is excluded and is blocked by the filter.

8. To view the configuration of a filter, use the following command:

   ```
   show log configuration filter {filter_name}
   ```

   The following is sample output from this command (for the earlier filter):

```
Log Filter Name: myFilter
I/                                          Severity
E  Comp.   Sub-comp.   Condition              CEWNISVD
-  ------- ----------- ---------------------- --------
I  STP     InBPDU                             --------
E  STP                 CreatPortMsgFail       -E------
I  STP                                        --------
Include/Exclude: I - Include,  E - Exclude
Component Unreg: * - Component/Subcomponent is not currently registered
Severity Values: C - Critical,  E - Error,  W - Warning,  N - Notice,  I - Info
Debug Severity : S - Debug-Summary,  V - Debug-Verbose,  D - Debug-Data
+ - Debug Severities, but log debug-mode not enabled
If Match parameters present:
Parameter Flags: S - Source,  D - Destination, (as applicable)
I - Ingress,  E - Egress,  B - BGP
Parameter Types: Port - Physical Port list,  Slot - Physical Slot #
MAC  - MAC address,  IP - IP Address/netmask,  Mask - Netmask
VID  - Virtual LAN ID (tag),  VLAN - Virtual LAN name
L4   - Layer-4 Port #,  Num  - Number,  Str  - String
Nbr  - Neighbor, Rtr  - Routerid, EAPS - EAPS Domain
Proc - Process Name
```

```
Strict Match   : Y - every match parameter entered must be present in the event
N - match parameters need not be present in the event
```

The `show log configuration filter` command shows each filter item, in the order that it will be applied and whether it will be included or excluded. The above output shows the three filter items, one including events from the STP.InBPDU component, one excluding the event STP.CreatPortMsgFail, and the next including the remaining events from the STP component. The severity value is shown as "*", indicating that the component's default severity threshold controls which messages are passed. The Parameter(s) heading is empty for this filter because no match is configured for this filter. Matches are described in Matching Expressions on page 1758.

Each time a filter item is added to or deleted from a given filter, the specified events are compared against the current configuration of the filter to try to logically simplify the configuration. Existing items will be replaced by logically simpler items if the new item enables rewriting the filter. If the new item is already included or excluded from the currently configured filter, the new item is not added to the filter.

Matching Expressions

You can configure the switch so messages reaching the target match a specified match expression.

The message text is compared with the configured match expression to determine whether to pass the message on. To require that messages match a match expression, use the following command:

`configure log target [`**`console`** | **`memory-buffer`** | **`nvram`** | **`primary-node`**| **`backp-node`** | **`session`** | **`syslog`** [**`all`** | *`ipaddress`* {**`udp-port`** {*`udp_port`*}} | *`ipPort`* | *`ipaddress`* **`tls-port`** {*`tls_port`*}] {**`vr`** *`vr_name`*} {**`local0 ... local7`**}] **`match`** [**`any`** |*`match-expression`*]

The messages reaching the target will match the match-expression, a simple regular expression. The formatted text string that makes up the message is compared with the match expression and is passed to the target if it matches. This command does not affect the filter in place for the target, so the match expression is compared only with the messages that have already passed the target's filter. For more information on controlling the format of the messages, see Formatting Event Messages on page 1761.

Simple Regular Expressions

A simple regular expression is a string of single characters including the dot character (.), which are optionally combined with quantifiers and constraints. A dot matches any single character, while other characters match only themselves (case is significant). Quantifiers include the star character (*) that matches zero or more occurrences of the immediately preceding token. Constraints include the caret character (^) that matches at the beginning of a message and the currency character ($) that matches at the end of a message. Bracket expressions are not supported. There are a number of sources available on the Internet and in various language references describing the

operation of regular expressions. The following table shows some examples of regular expressions.

Table 156: Simple Regular Expressions

Regular Expression	Matches	Does Not Match
port	port 2:3 import cars portable structure	poor por pot
..ar	baar bazaar rebar	bar
port.*vlan	port 2:3 in vlan test add ports to vlan port/vlan	
myvlan$	delete myvlan error in myvlan	myvlan port 2:3 ports 2:4,3:4 myvlan link down

Matching Parameters

Rather than using a text match, EMS allows you to filter more efficiently based on the parameter values of the message.

In addition to event components and conditions and severity levels, each filter item can also use parameter values to further limit which messages are passed or blocked. The process of creating, configuring, and using filters has already been described in Filtering By Components and Conditions on page 1756, so this section describes matching parameters with a filter item.

To configure a parameter match filter item, use the following command:

`configure log filter` *`name`* [**`add`** | **`delete`**] {**`exclude`**} **`events`** [*`event-condition`* | [**`all`** | *`event-component`*] {**`severity`** *`severity`* {**`only`**}}] [**`match`** | **`strict-match`**] *`type value`*

Each event in ExtremeXOS is defined with a message format and zero or more parameter types.

The `show log events all` command can be used to display event definitions (the event text and parameter types). Only those parameter types that are applicable given the events and severity specified are exposed on the CLI. The syntax for the parameter types (represented by *`type`* in the command syntax above) is:

[address-family [**`ipv4-multicast`** | **`ipv4-unicast`** | **`ipv6-multicast`** | **`ipv6-unicast`**] | **`bgp-neighbor`** *`ip address`* | **`bgp-routerid`** *`ip address`* | **`eaps`** *`eaps domain name`* | {**`destination`** | **`source`**} [**`ipaddress`** *`ip address`* | **`L4-port`** *`L4-port`*| **`mac-address`** *`mac-address`*] | **`esrp`** *`esrp domain name`* | {**`egress`** | **`ingress`**} [**`slot`** *`slot number`* | **`ports`** *`portlist`*] | **`ipaddress`** *`ip address`* | **`L4-port`** *`L4-port`* | **`mac-address`** *`mac_address`* | **`netmask`** *`netmask`* | **`number`**

number | **port** *portlist* | **process** *process name* | **slot** *slotid* | **string** *exact string to be matched* | **vlan** *vlan name* | **vlan tag** *vlan tag*]

You can specify the ipaddress type as IPv4 or IPv6, depending on the IP version.

The following examples show how to configure IPv4 addresses and IPv6 addresses:

IPv4 address

- To configure an IP address, with a mask of 32 assumed, use the following command:
  ```
  configure log filter myFilter add events all match ipaddress 12.0.0.1
  ```
- To configure a range of IP addresses with a mask of 8, use the following command:
  ```
  configure log filter myFilter add events all match ipaddress 12.0.0.0/8
  ```

IPv6 address

- To configure an IPv6 address, with a mask of 128 assumed, use the following command:
  ```
  configure log filter myFilter add events all match ipaddress 3ffe::1
  ```
- To configure a range of IPv6 addresses with a mask of 16, use the following command:
  ```
  configure log filter myFilter add events all match ipaddress 3ffe::/16
  ```

IPv6 scoped address

- IPv6 scoped addresses consist of an IPv6 address and a *VLAN*. The following examples identify a link local IPv6 address.

To configure a scoped IPv6 address, with a mask of 128 assumed, use the following command:

```
configure log filter myFilter add events all match ipaddress fe80::1%Default
```

To configure a range of scoped IPv6 addresses with a mask of 16, use the following command:

```
configure log filter myFilter add events all match ipaddress fe80::/16%Default
```

To configure a scoped IPv6 address with any VLAN, use the following command:

```
configure log filter myFilter add events all match ipaddress fe80::/16%*
```

To configure any scoped IPv6 address with a specific VLAN, use the following command:

```
configure log filter myFilter add events all match ipaddress fe80::/0%Default
```

Note
In the previous example, if you specify the VLAN name, it must be a full match; wild cards are not allowed.

The *value* depends on the parameter type specified.

As an example, an event may contain a physical port number, a source MAC address, and a destination MAC address. To allow only those *RADIUS (Remote Authentication*

Dial In User Service) incidents, of severity notice and above, with a specific source MAC address, use the following command:

```
configure log filter myFilter add events aaa.radius.requestInit severity notice match
source mac-address 00:01:30:23:C1:00
```

The string type is used to match a specific string value of an event parameter, such as a user name. The exact string is matched with the given parameter and no regular expression is supported.

Match Versus Strict-Match

The **match** and **strict-match** keywords control the filter behavior for those incidents with event definition that does not contain all the parameters specified in a `configure log filter events match` command.

This is best explained with an example. Suppose an event in the XYZ component, named XYZ.event5, contains a physical port number, a source MAC address, but no destination MAC address. If you configure a filter to match a source MAC address and a destination MAC address, XYZ.event5 will match the filter when the source MAC address matches regardless of the destination MAC address because the event contains no destination MAC address. If you specify the **strict-match** keyword, then the filter will never match event XYZ.event5 because this event does not contain the destination MAC address.

In other words, if the **match** keyword is specified, an incident will pass a filter so long as all parameter values in the incident match those in the match criteria, but all parameter types in the match criteria need not be present in the event definition.

Formatting Event Messages

Event messages are made up of a number of items. The individual items can be formatted; however, EMS does not allow you to vary the order of the items.

- Format the messages for a particular target.

  ```
  configure log target format
  ```

  Using the default format for the session target, an example log message might appear as:

  ```
  06/25/2004 22:49:10.63 <Info:dm.Info> MSM-A: PowerSupply:4 Powered On
  ```

  If you set the current session format using the following command:

  ```
  configure log target session format timestamp seconds date mm-dd-yyyy event-name
  component
  ```

  The same example would appear as:

  ```
  06/25/2004 22:49:10 <dm> PowerSupply:4 Powered On
  ```

- Provide some detailed information to technical support, set the current session format.

  ```
  configure log target session format timestamp hundredths date mmm-dd event-name
  condition process-name source-line
  ```

  The same example then appears as:

```
Jun 25 22:49:10.63 <dm.info> devmgr: (dm.c:134) PowerSupply:4 Powered On
```

Displaying Real-Time Log Messages

You can configure the system to maintain a running real-time display of log messages on the console display or on a (Telnet) session.

- Turn on the log display on the console with the command:

  ```
  enable log target console
  ```

  This setting may be saved to the FLASH configuration and is restored on boot-up (to the console display session).
- Turn on log display for the current session with the command:

  ```
  enable log target session
  ```

  This setting only affects the current session and is lost when you log off the session.

  The messages that are displayed depend on the configuration and format of the target. For information on message filtering, see Filtering Events Sent to Targets on page 1752. For information on message formatting, see Formatting Event Messages on page 1761.

Displaying Event Logs

The log stored in the memory buffer and the NVRAM can be displayed on the current session (either the console display or Telnet).

- Display the log using the command:

  ```
  show log {messages [memory-buffer | nvram]} {events {event-condition |
  event-component]} {severity severity {only}} {starting [date date time
  time | date date | time time]} {ending [date date time time | date
  date | time time]} {match regex} {chronological}
  ```

  You can use many options to select those log entries of interest. You can select to display only those messages that conform to the specified:

  - Severity
  - Starting and ending date and time
  - Match expression

  The displayed messages can be formatted differently from the format configured for the targets, and you can choose to display the messages in order of newest to oldest or in chronological order (oldest to newest).

Log Buffer Threshold Alert

ExtremeXOS 16.1 now supports the ability to generate syslog message or trigger a *SNMP* trap when the log memory buffer is filled up to 90% of configured log lines. SNMP/SNMP trap is not supported by ExtremeXOS EMS module at this point. The feature provides that an alert in the form of log message will be generated when the log memory buffer is filled up to a certain percentage threshold. A new command has

been added to allow you to configure the memory buffer alert. The alert will be based on the percentage used of the memory buffer. By default, the alert is disabled.

The new command can be used to enable the alert and to specify the percentage threshold that triggers alerts. When the percentage threshold is set to a value in the allowed range, the memory buffer alert is enabled. When the percentage threshold is set to none, the memory buffer alert is disabled. The alert triggered will be a log message with Notice severity.

If the alert is enabled, a log message will be generated in the following three scenarios:

1. When the actual use of the memory buffer first exceeds the configured percentage threshold.
2. When the configured memory buffer size is changed and the actual use of the buffer exceeds the configured percentage threshold.
3. When the configured percentage threshold is changed and the actual use of the buffer exceeds the re-configured percentage threshold.

When you execute the command `clear log`, the memory buffer will be emptied. When the new logs re-fill up to the configured percentage threshold, the log message will be re-generated. Output of `show log configuration` will be modified as well to include the information of memory buffer alert configuration and current memory buffer usage.

Uploading Event Logs

The log stored in the memory buffer and the NVRAM can be uploaded to a TFTP server.

- Use the following command to upload the log:

  `upload log [` *ipaddress* | *hostname*`]` {**vr** *vr_name*} {**block-size** *block_size*} *filename* {**messages** [**memory-buffer** | **nvram**] } {**severity** *severity* {**only**}} {**match** *regex*} {**chronological**}

  You must specify the TFTP host and the filename to use in uploading the log.

  There are many options you can use to select the log entries of interest. You can select to upload only those messages that conform to the specified:

  - Severity
  - Match expression

  The uploaded messages can be formatted differently from the format configured for the targets, and you can choose to upload the messages in order of newest to oldest or in chronological order (oldest to newest).

Displaying Counts of Event Occurrences

EMS adds the ability to count the number of occurrences of events. Even when an event is filtered from all log targets, the event is counted.

- Display the event counters.

  `show log counters {event condition | [all | event component]} {include | notified | occurred} {severity severity {only}}}`

  The system displays two counters. One counter displays the number of times an event has occurred, and the other displays the number of times that notification for the event was made to the system for further processing. Both counters reflect totals accumulated since reboot or since the counters were cleared using the `clear log counters` or `clear counters` command.

  The `show log counters` command also displays an included flag (the column titled In in the output). The included flag is set to Y(es) if one or more targets are receiving notifications of this event without regard to matching parameters.

  The keywords include, notified, and occurred display events only with non-zero counter values for the corresponding counter.

  The output of the command:

  ```
  show log counters stp.inbpdu severity debug-summary
  ```

  is similar to the following:

  ```
  Comp    SubComp     Condition               Severity      Occurred  In Notified
  ------- ----------- ----------------------- ------------- --------  -- --------
  STP     InBPDU      Drop                    Error                0  Y         0
  STP     InBPDU      Ign                     Debug-Summary        0  N         0
  STP     InBPDU      Mismatch                Warning              0  Y         0
  Occurred  : # of times this event has occurred since last clear or reboot
  Flags     : (*) Not all applications responded in time with there count values
  In(cluded): Set to Y(es) if one or more targets filter includes this event
  Notified  : # of times this event has occurred when 'Included' was Y(es)
  ```

  The output of the command:

  ```
  show log counters stp.inbpdu.drop
  ```

  is similar to the following:

  ```
  Comp    SubComp     Condition               Severity      Occurred  In Notified
  ------- ----------- ----------------------- ------------- --------  -- --------
  STP     InBPDU      Drop                    Error                0  Y         0
  Occurred  : # of times this event has occurred since last clear or reboot
  Flags     : (*) Not all applications responded in time with there count values
  In(cluded): Set to Y(es) if one or more targets filter includes this event
  Notified  : # of times this event has occurred when 'Included' was Y(es)
  ```

Displaying Debug Information

By default, a switch does not generate events of severity Debug-Summary, Debug-Verbose, and Debug-Data unless the switch is in debug mode. Debug mode causes a performance penalty, so it should only be enabled for specific cases where it is needed.

- Place the switch in debug mode using the command:

  ```
  enable log debug-mode
  ```

  When the switch is in debug-mode, any filters configured for your targets still affect which messages are passed on or blocked.

Logging Configuration Changes

ExtremeXOS allows you to record all configuration changes and their sources that are made using the CLI by way of Telnet or the local console. These configuration changes are noted in the log as event entries. Each log entry includes the user account name that performed the change and the source IP address of the client (if Telnet was used). Configuration logging applies only to commands that result in a configuration change. CLI configuration logging is disabled by default.

- To enable configuration logging, use the following command:

  ```
  enable cli config-logging
  ```

- To d configuration logging, use the following command:

  ```
  disable cli config-logging
  ```

You can specify that logged commands appear in fully expanded form, rather than the abbreviations you may have used when entering them in the command line.

For example, with command expansion enabled, a command entered in abbreviated format, such as

```
config por 33 auto of spee 10000 duplex ful
```

appears in the log as:

```
configure ports 33 auto off speed 10000 duplex full
```

- To enable command expansion, use the following command:

  ```
  enable cli-config-logging expansion
  ```

- To disable command expansion, use the following command:

  ```
  disable cli-config-logging expansion
  ```

Securing Syslog Transport Layer Security Sessions

By default, the following ciphers are enabled for Syslog Transport Layer Security (TLS) sessions:

aes128-sha, aes128-sha256; aes256-sha256; dhe-rsa-aes128-sha256; dhe-rsa-aes256-sha256

You can selectively enable and disable ciphers using the following command:

```
configure syslog tls cipher [[cipher | all] on | cipher off]
```

To see which ciphers are enabled or disabled for Syslog TLS sessions, use the following command:

```
show log configuration
```

Avoiding Potential Loss of TLS Syslog Logging

For Linux, by default, it takes about 15 minutes for kernel to end a TCP connection when transmitted data remains unacknowledged. This results in a potential loss of logs to TLS Syslog server during the 15 minutes window due to link down.

To reduce this time window, use the following command:

```
configure syslog tls tcp-user-timeout [seconds | default]
```

To view the value set for Syslog TLS TCP user timeout, use the following comamnd:

```
show log configuration
```

Disabling OCSP for TLS Connections to Remote Syslog Servers

For compliance with *RFC 6960* (X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP), you can disable the OCSP check for Transport Layer Security (TLS) connections to remote Syslog servers by using the following command:

```
configure syslog tls ocsp [on | off]
```

To see the status of OCSP checking, use the following command:

```
show log configuration
```

Note
Be sure you understand the ramifications of turning off OCSP if you chose to do so.

Syslog TLS OCSP Attribute Configuration

Beginning with version 32.2, you can configure Syslog TLS OCSP attributes (nonce, override, and ocsp-nocheck, respectively) using the following commands:

- `configure syslog tls ocsp nonce [on | off]`
- `configure syslog tls tls override [url | none]`
- `configure syslog tls ocsp signer ocsp-nocheck [on | off]`

OCSP nonce cryptographically binds an OCSP request and an OCSP response with an `id-pkix-ocsp-nonce` extension to prevent replay attacks.

OCSP override configures one HTTP Online Certificate Status Protocol (OCSP) override URL for TLS connections to a remote Syslog server.

OCSP-nocheck

If this configuration is turned on, then the operating system assumes the OCSP signer certificate contains the ocsp-nocheck extension, whether it has the extension or not. This results in the system accepting the OCSP responses from this OCSP signer. By default, this configuration is off to maintain backward compatibility, but can be turned on or off per application. This is also applicable for both the override server and AIA servers.

Using the XML Notification Client

Introduction

This feature allows an event such as a configuration change, a fault, a change in status, the crossing of a threshold, or an external input to the system, to be sent as an asynchronous message or event notification to external web servers.

The only ExtremeXOS modules that support XML notification as targets are Identity Management and EMS.

XML notification does not provide any event filtering capability. However, in the case of EMS, the target Web server can be configured with log filters and conditions. The XML notification feature establishes and checks the connectivity with the web server only when an event is ready to be pushed. State transitions take place if required. Statistics are updated accordingly and can be monitored.

The Web servers must be configured and enabled using ExtremeXOS CLI with an IP address, port, protocol type, user authentication, session information, if any, and other web server configuration.

A maximum of four web servers can be configured at a time.

The XML schemas are defined using Web Services Description Language (WSDL) in the XML SDK.

HTTP Client Interface

The event notifications are sent in XML format, using SOAP transport protocol and HTTP/HTTPS.

The XML notification client can communicate with the external HTTP/HTTPS server. HTTP/HTTPS client interfaces maintain persistent connections with external Web servers as long as the target is activated on the switch.

The HTTP URL format for the server is `http://<ip-address>:<port>/<service>`. The default HTTP port number 80 is used if a port number is not configured.

HTTP basic access authentication method (Base64 algorithm) is used to encrypt the user name and password when making a request. HTTP cookies are not supported in this release.

Create a server certificate that can be used by the HTTPS server. To generate the secure certificate on a ExtremeXOS switch, see the configuration guidelines of the HTTP server, (see Secure Socket Layer on page 1676 for more information).

Configuring XML Notification

- Create a web server target on an XML client using the command:

  `create xml-notification target new-target **url** url {**vr** vr_name} {**user** [**none** | user]} {**encrypted-auth** encrypted-auth} {**queue-size** queue-size}`
- Configure a web server target on an XML client using the command:

  `configure xml-notification target target [**url** url {**vr** vr_name} | **user** [**none** | user] | [**encrypted-auth** encrypted-auth] | [**queue-size** queue-size]]`
- Add or delete an ExtremeXOS application to a web server target (EMS or Identity Management) using the command:

  `configure xml-notification target target [**add** | **delete**] module`
- Enable or disable web server target(s) using the command:

  `[**enable**|**disable**] **xml-notification** [**all** | target]`
- Delete a web server target on an XML client process using the command:

  `delete xml-notification target target`
- Unconfigure an XML client process using the command:

  `unconfigure xml-notification`
- Unconfigure and reset all statistics counters using the command:

  `clear counters xml-notification {**all** | target}`

Displaying XML Notification

- Display the configuration of a web server target using the command:

  `show xml-notification configuration {target}`
- Display the connection status, enable status, and event statistics of the target web server using the command:

  `show xml-notification statistics {target}`
- Display information on the stored certificate using the command:

  `show ssl {**detail**}`

Configuring Log Target in EMS

The following commands support the EMS XML target.

- Create a web server XML target using hte command:
  `create log target xml-notification [ target_name | xml_target_name ]`
- Configure the web server target with an EMS filter using the command:
  `configure log target xml-notification xml_target_name` **`filter`** *`filter-name`* {**`severity`** [[*`severity`*] {**`only`**}]}
- Enable the web server target using the command:
  `enable log target xml-notification xml_target_name`
- Disable the web server target using the command:
  `disable log target xml-notification xml_target_name`
- Delete the web server target XML target using the command:
  `delete log target xml-notification xml_target_name`
- Display XML target information using the command:
  `show log configuration target xml-notification {xml_target_name}`

Examples

Below are examples of configuring web server targets in a XML Notification module.

Scenario 1: Push filtered EMS events to external web server in a well-defined XML format.

Create a Web Server Target test1, create a log target and a filter in EMS, and attach the filter to the web target. Enable the target in both EMS and XML-Notification module.

```
create XML-notification target test1 url http://10.255.129.22:8080/xos/webservice user
admin
create log target xml-notification "test1"
create log filter xmlc_filter_1
configure log filter "xmlc_filter_1" add events idmgr
configure log target xml-notification "test1" filter "xmlc_filter-1"
enable log target xml-notification "test1"
enable XML-notification test1
```

Scenario 2: Push user identity events to the external web server without EMS module in a well defined (XML Schema) XML format.

Create a web server target and attach an idmgr module. Idmgr modules use an XML-notification backend library to trigger events. In this case, no special filters are supported.

```
create xml-notification target test2 url http://10.255.129.22:8080/xos/webservice user
admin
configure xml-notification target test2 add module idmgr
enable xml-notification test2
```

Scenario - 3: XMLC notifications using HTTPS

Install the SSH module. If not installed, refer to Understanding System Redundancy on page 84 for details on SSH and Secure Socket Layer on page 1676 for details on SSL.

```
configure ssl certificate privkeylen 1024 country us organization extreme common-name
name1
```

```
create xml-notification target test3 url https://10.120.91.64:8443/xos/webservice
configure xml-notification target "test3" user admin
configure xml-notification target "test3" add "ems"
enable xml-notification "test3"
```

Using sFlow

sFlow is a technology for monitoring traffic in data networks containing switches and routers. It relies on statistical sampling of packets from high-speed networks, plus periodic gathering of the statistics. A *UDP (User Datagram Protocol)* datagram format is defined to send the information to an external entity for analysis. sFlow consists of a *MIB (Management Information Base)*, and a specification of the packet format for forwarding information to a remote agent. For details on sFlow specifications, see RFC 3176; and for more general information, go to www.sflow.org.

The ExtremeXOS implementation is based on sFlow version 5, which is an improvement from the revision specified in RFC 3176.

Additionally, the switch software also allows you to set the individual port sampling rates, so that you can fine-tune sFlow statistics.

sFlow and mirroring are not mutually exclusive on ExtremeSwitching series switches, whether or not they are included in a SummitStack. You can enable them simultaneously.

However, be aware that the following limitations are present in the ExtremeXOS implementation:

- Generic port statistics are reported to the sFlow collector
- Non-extended data
- Only port-based sampling
- No MIB support

Sampling Mechanisms

ExtremeSwitching series switches support hardware-based sampling at a programmed interval.

With hardware-based sampling, the data path for a packet that traverses the switch does not require processing by the CPU. Fast path packets are handled entirely by ASICs and are forwarded at wire speed rate.

Both ingress and egress sFlow sampling can be enabled simultaneously on a port. The **enable sflow port** command provides an option to enable sFlow on ingress, or egress, or both directions. The default value is ingress. The sample rate is maintained on a per-port basis, so a given port has the same sample rate for ingress and egress traffic. Ingress and egress sFlows sample both unicast and multicast egress flows. The global enable/disable control of sFlow is common to both ingress and egress. When the global option is enabled, the port level sFlow parameter is applied to hardware.

When sFlow sampling is enabled on a port, the sFlow agent samples the traffic on that port, processed in slow path and passed on to the collector. You can configure the rate at which the packets are sampled.

ExtremeXOS 22.5 expands upon sFlow's capability by providing support for additional data structures that an sFlow agent can use to report table utilization statistics in sFlow counter samples (see Displaying sFlow Information on page 1777).

Limitations of Egress sFlow

The following limitations apply to the egress sFlow feature:

- Due to the hardware limitation, destination port information is not supported for multicast traffic. The output interface index is populated as 0.
- Egress sFlow sampling does not support de-duplication of packets.
- For multicast traffic, the sampling rate, sample pool of the egress sFlow sampled datagram is populated as 0, because the source ID of the egress sampled multicast packet is unknown.
- For L3 unicast traffic, an unmodified packet is sampled and the destination port is supplied if the L3 traffic is a flow within single chip. When the egress port and ingress port are in different chips, then a modified packet is sampled and the destination ports are supplied. For L3 multicast traffic, unmodified packet is sampled and destination port is populated as zero.
- Packets dropped due to egress *ACL* are sampled.
- In cases of unicast and multicast flooding, the packets are sampled before packet replication. If the ingress and member ports are in the same chip, then a single copy of the packet is sampled even though the egress sFlow is enabled on more than one member's ports. If the member ports are spread across different chip, then packets are sampled on a per-chip basis.
- In flooding cases, the least configured sampling rate among the member ports on a port group is considered as a sample rate. Even if you configure different sample rates on a member ports, egress sampling is based on the least configured sample rate among the member ports on a chip.

sFlow Destination Port

Currently, when a packet is sampled from the ingress traffic, the output interface index is populated as 0. This is applicable for both unicast and multicast traffic. As part of the egress sFlow, the output interface index is populated with the destination port information for unicast packets sampled in both ingress and egress directions. Multicast packet samples still have an output interface index as 0.

Table 157 illustrates the expected output of the ifIndex when based on traffic type, and sampling direction.

Table 157: IfIndex Traffic Type Sampling Direction

Case #	Ingress/Engress Unicast/ Multicast	Scenario
1	Ingress/unicast	sFlow sample includes both ingress and egress port (ifindex).
2	Egress/unicast	sFlow sample includes both ingress and egress port information.
3	Ingress/multicast	Egress port information cannot be provided because of hardware limitation. Egress ifindex is 0 and ingress ifindex is supplied.
4	Egress/multicast	Egress port information cannot be provided because of hardware limitation. Egress ifindex is 0 and ingress ifindex is supplied.

Configuring sFlow

ExtremeXOS allows you to collect sFlow statistics on a per port basis.

An agent residing locally on the switch sends data to a collector that resides on another machine. You configure the local agent, the address of the remote collector, and the ports of interest for sFlow statistics gathering. You can also modify default values for how frequently on an average a sample is taken and the maximum number of samples allowed before throttling the sample gathering.

To configure sFlow on a switch:

- Configure the local agent
- Configure the addresses of the remote collectors
- Enable sFlow globally on the switch
- Enable sFlow on the desired ports

Optionally, you may also change the default values of the following items:

- How often the statistics are collected
- How frequently a sample is taken, globally or per port
- How many samples per second can be sent to the CPU

Configuration Tasks

Use the following commands to configure the sFlow feature:

- `enable sflow ports` **`all`** | *`port_list`* {**`ingress`** | **`egress`** | **`both`**}

The keyword options enable you to configure sFlow types on a given set of ports. If you do not configure an sFlow type, then ingress sFlow sampling is enabled as the default configuration.

Use the following commands to display the type of sFlow configured on the physical interface, and various statistics about sFlow sampling:

- `show sflow configuration`
- `show sflow statistics`

  The following fields are displayed:
  - Received frames—Number of frames received on sFlow enabled ports.
  - Sampled Frames—Number of packets that have been sampled by sFlow.
  - Transmitted Frames—Number of UDP packets sent to remote collector(s).
  - Broadcast Frames—Number of broadcast frames received on sFlow enabled ports.
  - Multicast Frames—Number of multicast frames received on sFlow enabled ports.
  - Packet Drops—Number of samples dropped.

Configuring the Local Agent

The local agent is responsible for collecting the data from the samplers and sending that data to the remote collector as a series of UDP datagrams. The agent address is stored in the payload of the sFlow data, and is used by the sFlow collector to identify each agent uniquely. By default, the agent uses the management port IP address as its IP address.

- Change the agent IP address using the command:
  `configure sflow agent {`**`ipaddress`**`} ipaddress`
- Unconfigure the agent using the command:
  `unconfigure sflow agent`

Configuring the Remote Collector Address

You can specify up to four remote collectors to send the sFlow data to. Typically, you would configure the IP address of each collector. You may also specify a UDP port number different from the default value of 6343, and/or a different from the default of *VR-Mgmt*. When you configure a collector, the system creates a database entry for that collector that remains until the collector is unconfigured. All the configured collectors are displayed in the command: `show sflow {`**`configuration`**`}`.

- Configure the remote collector using the command:
  `configure sflow collector {`**`ipaddress`**`} ipaddress {`**`port`** `udp-port-number}` `{`**`vr`** `vr_name}`
- Unconfigure the remote collector and remove it from the database using the command:
  `unconfigure sflow collector {`**`ipaddress`**`} ipaddress {`**`port`** `udp-port-number} {`**`vr`** `vr_name}`

Enabling sFlow Globally on the Switch

Before the switch starts sampling packets for sFlow, you must enable sFlow globally on the switch.

- Enable sFlow globally using the command:

  ```
  enable sflow
  ```

- Disable sFlow globally using the command:

  ```
  disable sflow
  ```

  When you disable sFlow globally, the individual ports are also put into the disabled state. If you later enable the global sFlow state, individual ports return to their previous state.

Enabling sFlow on the Desired Ports

- Enable sFlow on specific ports using the command:

  ```
  enable sflow ports port_list {ingress | egress | both}
  ```

  The ingress option allows you to configure the sFlow type on a given set of ports. This option is configured on the port by default.

  You may enable and disable sFlow on ports irrespective of the global state of sFlow, but samples are not taken until both the port state and the global state are enabled.

- Disable sFlow on ports using the command:

  ```
  disable sflow ports port_list
  ```

Additional sFlow Configuration Options

You can configure three global options to different values from the defaults. These options affect how frequently the sFlow data is sent to the remote collector, how frequently packets are sampled, and the maximum number of sFlow samples that could be processed in the CPU per second.

You can also configure how frequently packets are sampled per port.

Polling Interval

Each port counter is periodically polled to gather the statistics to send to the collector. If there is more than one counter to be polled, the polling is distributed in such a way that each counter is visited once during each polling interval, and the data flows are spaced in time. For example, assume that the polling interval is 20 seconds and there are 40 counters to poll. Two ports will be polled each second, until all 40 are polled.

- Configure the polling interval using the command:

  ```
  configure sflow poll-interval seconds
  ```

Global Sampling Rate

The global sampling rate is the rate set on newly enabled sFlow ports. Changing this rate does not affect currently enabled sFlow ports. The default sample rate is 8192, so sFlow samples one packet out of every 8192 received.

- Use the following command to configure the switch to use a different sampling rate.

  ```
  configure sflow sample-rate number
  ```

For example, if you set the sample rate number to 16384, the switch samples one out of every 16384 packets received. Higher numbers mean fewer samples and longer times between samples. If you set the number too low, the number of samples can be very large, which increases the load on the switch. Do not configure the sample rate to a number lower than the default unless you are sure that the traffic rate on the source is low.

The Global Sampling Rate applies to all ExtremeSwitching platforms.

Note
The minimum rate that these platforms sample is 1 out of every 256 packets. If you configure a rate to be less than 256, the switch automatically rounds up the sample rate to 256.

Per Port Sampling Rate

The per port sampling rate overrides the system-wide value.

The rate is rounded off to the next power of two, so if 400 is specified, the sample rate is configured as 512. The valid range is [256-536870912].

- Set the sampling rate on individual ports.

  ```
  configure sflow ports port_list sample-rate number
  ```

Maximum CPU Sample Limit

A high number of samples can cause a heavy load on the switch CPU. To limit the load, there is a CPU throttling mechanism to protect the switch.

Note
Sflow sampling is limited to 2000 packets per second in the CPU. Any packets sent at a rate greater than 2000 pps are dropped.

On a stand-alone switch, whenever the limit is reached, the sample rate value is doubled on the ports from which the maximum number of samples are received. For ports that are sampled less frequently, the sampling rate is not changed; the sub-sampling factor is adjusted downward.

- Configure the maximum CPU sample limit.

  ```
  configure sflow max-cpu-sample-limit rate
  ```

Unconfiguring sFlow

- Reset the configured values for sFlow to their default values and remove from sFlow any configured collectors and ports using the following ocmmand:

  ```
  unconfigure sflow
  ```

sFlow Configuration Example

In a service provider environment, you can configure sFlow to sample packets at the edge of the network to determine the hourly usage for each IP address in the data center.

You can capture web traffic, FTP traffic, mail traffic, and all bits of data that travel across service providers' edge routers to their customers' (end users') servers.

The example in this section assumes that you already have an sFlow data collector installed somewhere in your network. In many environments, the sFlow data collector is on a network PC.

The following sFlow configuration example performs the following tasks in a service provider environment:

- Configures the IP address of the sFlow data collector.

  Note
  In many environments, the sFlow data collector is not directly connected to the switch. Make sure to specify the VR used to forward traffic between the sFlow collector and the switch. In most cases the VR is *VR-Mgmt*.

- Configures the sampling rate on an edge port.
- Enables sFlow on the edge port.
- Enables sFlow globally on the switch.

```
configure sflow collector 10.127.11.88 vr vr-mgmt
configure sflow ports 5:21 sample-rate 8192
enable sflow ports 5:21 egress
enable sflow
```

Here is sample output for the configuration:

```
SFLOW Global Configuration
Global Status: enabled
Polling interval: 20
Sampling rate: 8192
Maximum cpu sample limit: 2000
SFLOW Configured Agent IP: 0.0.0.0
Operational Agent IP: 10.127.11.88
Collectors
SFLOW Port  Configuration Port Status  Sample-rate Subsampling Sflow-type
Config / Actual factor Ingress/Egress
5:21    enabled          8192 / 8192  1          Disabled / Enabled
```

Displaying sFlow Information

- To display the current configuration of sFlow, use the following command:

  ```
  show sflow {configuration}
  ```

- To display the sFlow statistics, use the following command:

  ```
  show sflow statistics
  ```

- To display the sFlow hardware table utilization statistics, use the following command:

  ```
  show sflow hardware-utilization {slot [slot_num | all]}
  ```

Using RMON

Using the Remote Monitoring (RMON) capabilities of the switch allows network administrators to improve system efficiency and reduce the load on the network.

Note
You can use the RMON features of the system only if you have an RMON management application and have enabled RMON on the switch.

About RMON

RMON is the common abbreviation for the Remote Monitoring Management Information Base (MIB) system defined by the Internet Engineering Task Force (IETF) documents RFC 1757 and RFC 2021, which allows you to monitor LANs remotely.

A typical RMON setup consists of the following two components:

- RMON agent
- Management workstation

RMON Agent

An RMON agent is an intelligent software agent that continually monitors port statistics and system variables.

The agent transfers the information to a management workstation on request, or when a predefined threshold is crossed.

Information collected by RMON includes Ethernet port statistics and history and the software version and hardware revision of the device. RMON generates alarms when threshold levels are met and then logs those events to the log. RMON can also send traps to the destination address configured by the management workstation. You can also use RMON to trigger a system reboot.

Management Workstation

A management workstation communicates with the RMON agent and collects the statistics from it.

The workstation does not have to be on the same network as the RMON agent and can manage the agent by in-band or out-of-band connections.

If you enable RMON on the switch, you can use a management workstation to review port statistics and port history, no configuration of the management workstation is necessary. However, you must use a management workstation to configure the alarm and event entries.

Supported RMON Groups of the Switch

The IETF defines nine groups of Ethernet RMON statistics. The switch supports the following four of these groups, as defined in RFC 1757:

- Statistics
- History
- Events

The switch also supports the following parameters for configuring the RMON agent and the trap destination table, as defined in RFC 2021:

- probeCapabilities
- probeSoftwareRev
- probeHardwareRev
- probeDateTime
- probeResetControl
- trapDestTable

The following sections describe the supported groups, the RMON probe configuration parameters, and the trap destination parameter in greater detail.

Statistics

Per virtual network and per remote endpoint counters are supported. Both byte and packet counts are available.

Virtual network counters display the sum of all counters from all remote endpoints associated with the virtual network. Packets that arrived encapsulated on the VXLAN network port are counted on the virtual network as Rx bytes/packets (network side), and also, are counted as Tx bytes/packets (access side). For example, if 10 packets are received from remote endpoint 1.1.1.2 on virtual network vnet44, Rx Frames are 10, since 10 frames are received on the virtual network from the network interface (VXLAN encapsulated packets). Also, TX frames are 10, since 10 frames are transmitted on the virtual network tenant interface (native, non-VXLAN encapsulated packets). Similarly, on the other direction.

Remote endpoint counters display the sum of all counters from all virtual networks associated with the remote endpoint. Currently, it is not possible to display the counters for a given virtual network/remote endpoint pair.

History

The History group provides historical views of network performance by taking periodic samples of the counters supplied by the Statistics group.

The group features user-defined sample intervals and bucket counters for complete customization of trend analysis.

The group is useful for analysis of traffic patterns and trends on an Ethernet port, and for establishing baseline information indicating normal operating parameters.

Events

The Events group creates entries in an event log and/or sends *SNMP* traps to the management workstation.

An event is triggered by an RMON alarm. The action taken can be configured to ignore it, to log the event, to send an SNMP trap to the receivers listed in the trap receiver table, or to both log and send a trap. The RMON traps are defined in RFC 1757 for rising and falling thresholds.

Effective use of the Events group saves you time. Rather than having to watch real-time graphs for important occurrences, you can depend on the Events group for notification. Through the SNMP traps, events can trigger other actions, which provides a mechanism for an automated response to certain occurrences.

RMON Probe Configuration Parameters

The RMON probe configuration parameters supported in ExtremeXOS are a subset of the probe configuration group as defined in RFC 2021. The probe configuration group controls and defines the operation of the RMON agent.

You can configure the following objects:

- probeCapabilities—View the RMON MIB groups supported on at least one interface by the probe.
- probeSoftwareRev—View the current software version of the monitored device.
- probeHardwareRev—View the current hardware version of the monitored device.
- probeDateTime—View the current date and time of the probe. For example, Friday December 31, 2004 at 1:30:15 PM EST is displayed as: 2004-12-31,13:30:15.0

  If the probe is aware of time zones, the display also includes the Greenwich Mean Time (GMT) offset. For example, Friday, December 31, 2004, 1:30:15 PM EST with the offset known is displayed as: 2004-12-31,13:30:15.0, -4.0

  If time information is unavailable or unknown, the time is not displayed.
- probeResetControl—Restart a managed device that is not running normally. Depending on your configuration, you can do one of the following:
  - Warm boot—A warm boot restarts the device using the current configuration saved in non-volatile memory.
  - Cold boot—A cold boot causes the device to reset the configuration parameters stored in non-volatile memory to the factory defaults and then restarts the device using the restored factory default configuration.

trapDestTable

The trapDestTable contains information about the configured trap receivers on the switch and stores this information in non-volatile memory.

To configure one or more trap receivers, see Using the Simple Network Management Protocol on page 106.

Extreme-RtStats-MIB

The extremeRtStatsTable provides the user with all the common measurement/monitoring attributes in a single table.

It includes measurements like utilization, error, and collision levels.

The extreme RtStatsUtilization variable gives an accurate measurement of segment utilization. ExtremeRtStatsCollisions is included in the segment utilization calculation for more accuracy. Collision statistics are collected periodically, and the segment utilization is calculated with a sampling interval of 5 minutes (300 seconds).

The extremeRtStatsTotalErrors variable is calculated by adding the following counters:

- extremeRtStatsCRCAlignErrors (receive errors)
- extremeRtStatsFragments (receive errors)
- extremeRtStatsJabbers (receive errors)
- extremeRtStatsCollisions (transmit errors)

Configuring RMON

RMON requires one probe per LAN segment, and stand-alone RMON probes traditionally have been expensive. Therefore, the approach taken by Extreme Networks has been to build an inexpensive RMON probe into the agent of each system. This allows RMON to be widely deployed around the network without costing more than traditional network management. The switch accurately maintains RMON statistics at the maximum line rate of all of its ports.

By default, RMON is disabled. However, even in the disabled state, the switch collects etherStats and you can configure alarms and events.

RMON saves the history, alarm, and event configurations to the configuration file. Runtime data is not stored in the configuration file and is subsequently lost after a system restart.

- Enable or disable the collection of RMON statistics on the switch using the commands:

  ```
  enable rmon
  ```

  ```
  disable rmon
  ```

By enabling RMON, the switch begins the processes necessary for collecting switch statistics.

Event Actions

The actions that you can define for each alarm are shown in Table 158.

Table 158: Event Actions

Action	High Threshold
no action	
log	Sends a log message.
log-and-trap	Sends both a log message and a trap to all trap receivers.
snmp-trap	Sends a trap to all trap receivers.

To be notified of events using *SNMP* traps, you must configure one or more trap receivers, as described in Using the Simple Network Management Protocol on page 106.

Display RMON Information

- View the status of RMON polling on the switch (the enable/disable state for RMON polling) using the command:

  ```
  show switch management
  ```

- View the RMON memory usage statistics for a specific RMON feature (for example, statistics, events, logs, or history) or for all features using the command:

  ```
  show rmon memory {detail | memoryType}
  ```

SMON

SMON is the common abbreviation for the Switch Network Monitoring Management Information Base (MIB) system defined by the Internet Engineering Task Force (IETF) document RFC 2613.

SMON is a set of MIB extensions for RMON that allows monitoring of switching equipment from a *SNMP* Manager in greater detail. The supported MIB tables are described in Supported Standards, Protocols, and MIBs on page 1854; smonPrioStatsControlTable and smonPrioStatsTable cannot be supported due to hardware limitations.

Note

When you delete all the mirroring filters through the portCopyConfigTable, the mirroring is disabled automatically.

Monitoring CPU Utilization

You can monitor the CPU utilization and history for all of the processes running on the switch.

By viewing this history on a regular basis, you can see trends emerging and identify processes with peak utilization. Monitoring the workload of the CPU allows you to troubleshoot and identify suspect processes before they become a problem. By default, the switch monitors CPU utilization every five seconds. In addition, when CPU utilization of a process exceeds 90% of the regular operating basis, the switch logs an error message specifying the process name and the current CPU utilization for the process.

Disabling CPU Monitoring

- Disable CPU monitoring using the command:

  ```
  disable cpu-monitoring
  ```

This command disables CPU monitoring on the switch; it does not clear the monitoring interval. Therefore, if you altered the monitoring interval, this command does not return the monitoring interval to five seconds. The next time you enable CPU monitoring, the switch uses the existing configured interval.

Enable CPU Monitoring

By default, CPU monitoring is enabled and occurs every five seconds. The default CPU threshold value is 90%.

- Enable CPU monitoring using the command:

  ```
  enable cpu-monitoring {interval seconds} {threshold percent}
  ```

  Where the following is true:

  - *seconds*—Specifies the monitoring interval. The default interval is 5 seconds, and the range is 5 to 60 seconds. We recommend the default setting for most network environments.
  - threshold—Specifies the CPU threshold value. CPU usage is measured in percentages. The default is 90%, and the range is 0% to 100%.

Displaying CPU Utilization History

- Display the CPU utilization history of one or more processes using the command:

  ```
  show cpu-monitoring {process name} {slot slotid}
  ```

  Where the following is true:

  - name—Specifies the name of the process.
  - slot—On a SummitStack, specifies the slot number of the target node. The number is a value from one (1) to eight (8).

  Output from this command includes the following information:

  - Process—The name of the process.
  - Range of time (five seconds, ten seconds, and so forth)—The CPU utilization history of the process or the system. The CPU utilization history goes back only 1 hour.

  - Total User/System CPU Usage—The amount of time recorded in seconds that the process spends occupying CPU resources. The values are cumulative meaning that the values are displayed as long as the system is running. You can use this information for debugging purposes to see where the process spends the most amount of time: user context or system context.

- Clear the utilization history stored in the switch and reset the statistics to zero using the command:

  ```
  clear cpu-monitoring {process name} {slot slotid}
  ```

The following is sample truncated output from a switch:

```
5420F-48P-4XE.1 # show cpu-monitoring

      CPU Utilization Statistics - Monitored every 5 seconds
-------------------------------------------------------------------------

Process      5   10   30   1    5    30   1    Max          Total
            secs secs secs min  mins mins hour           User/System
            util util util util util util util util       CPU Usage
            (%)  (%)  (%)  (%)   (%)  (%)  (%)  (%)         (secs)
-------------------------------------------------------------------------

System        3.7  3.7  3.6  3.6  3.5  3.5  3.3  5.3    17.63   158657.42
aaa           0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0    15.55       18.87
acl           0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.6   138.28      214.65
bfd           0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.6    75.59       62.81
bgp           0.0  0.0  0.0  0.0  0.0  0.0  0.0  2.1     0.25        0.07
brm           0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3     0.06        0.03
cfgmgr        0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.2     0.46        0.23
cli           0.0  0.1  0.0  0.0  0.0  0.0  0.0 16.0    10.05        3.42
devmgr        0.0  0.0  0.0  0.0  0.0  0.0  0.0  3.5     0.60        0.33
dirser        0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.3     0.15        0.22
dosprotect    0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3     0.09        0.06
dot1ag        0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0     1.58        2.23
```

Non-Volatile Storage Health Monitoring

Version 33.6.1 adds health monitoring capabilities for non-volatile (flash) storage on Summit ARM platforms (4220, 5320, 5420, 5520 series). The system automatically monitors flash memory health and logs diagnostic information based on configurable reporting intervals.

Health Monitoring Overview

Flash memory has a finite lifespan measured in erase cycles. Over time, blocks in the flash memory can become unusable due to wear. The health monitoring feature tracks key metrics that indicate the current state of the flash memory:

- **Erase counter:** Total number of erase cycles performed on the flash device
- **Bad blocks:** Blocks that are no longer usable
- **Reserved blocks:** Blocks reserved for wear leveling and bad block replacement
- **Available blocks:** Blocks currently available for use
- **Total blocks:** Total number of blocks in the flash device

Monitoring runs automatically by default and cannot be disabled. This ensures potential storage issues are detected before they impact switch operation.

Viewing Health Status

To view the current health status of non-volatile storage, use the following command:

```
show switch non-volatile-storage health-check
```

Sample output:

```
Poll period mode: normal
Erase counter:       134
Block counts:
 Bad:                  6
 Reserved:            74
 Available:          969
 Total:             4090
```

The output provides a snapshot of the current flash memory state. If values indicate potential issues, contact Extreme Networks Technical Support for evaluation and recommendations.

Configuring Health Check Reporting

You can configure how frequently the system checks and reports on storage health. Two reporting modes are available:

Normal Mode (Default):

In normal mode, the system performs health checks every 24 hours and logs messages only when values indicate potential issues that should be investigated by Technical Support. This mode provides adequate monitoring while minimizing log verbosity.

```
configure switch non-volatile-storage health-check-reporting normal
```

Frequent Mode:

In frequent mode, the system performs health checks every hour and logs all values regardless of whether operation is nominal. Use this mode when:

- Closely monitoring a switch suspected of having storage issues
- Gathering detailed diagnostic data for Technical Support analysis
- Validating storage health after maintenance operations

```
configure switch non-volatile-storage health-check-reporting frequent
```

Note
Health check configuration persists across reboots.

Interpreting Health Metrics

Understanding the health metrics helps you assess the current state of your flash storage:

Erase Counter:

The erase counter indicates how many erase operations have been performed. Flash memory has a limited number of erase cycles (typically tens of thousands to hundreds of thousands depending on the memory type). A steadily increasing erase counter is normal during switch operation. Rapid increases may indicate excessive write activity.

Bad Blocks:

Bad blocks are regions of flash memory that can no longer reliably store data. A small number of bad blocks (typically less than 1% of total blocks) is normal for flash memory. The system automatically avoids using bad blocks. A rapidly increasing bad block count may indicate accelerated wear or hardware issues.

Reserved Blocks:

Reserved blocks are set aside for wear leveling and bad block replacement. The flash controller uses these blocks to replace bad blocks as they develop, maintaining available capacity. This is a normal part of flash memory management.

Available Blocks:

Available blocks are currently usable for data storage. As bad blocks develop, available blocks decrease while bad blocks increase. If available blocks drop significantly (more than 5-10% reduction from initial capacity), contact Technical Support.

When to Contact Support:

- Available blocks have decreased by more than 10% from the initial capacity
- Bad blocks are increasing rapidly (more than 5-10 blocks per month)
- System logs indicate storage health concerns
- The erase counter shows unusually rapid growth
- You observe unexpected system behavior that could be storage-related

Log Messages

In normal mode, the system logs messages only when health metrics indicate values that warrant Technical Support investigation. In frequent mode, the system logs all metrics on the configured schedule even when operation is nominal.

Log messages appear in the system log (`/var/log/messages`) and can be viewed using standard log viewing commands.

Best Practices

- **Monitor regularly:** Periodically check storage health status, especially on switches in high-write environments
- **Use normal mode:** Keep reporting in normal mode unless actively troubleshooting to avoid excessive log growth

- **Document baseline:** Record initial health metrics when deploying switches to establish baseline values
- **Contact support early:** If you observe concerning trends in health metrics, contact Technical Support before issues become critical
- **Plan replacements:** If Technical Support identifies storage degradation, plan for hardware replacement during the next maintenance window

Debug Testing Mode

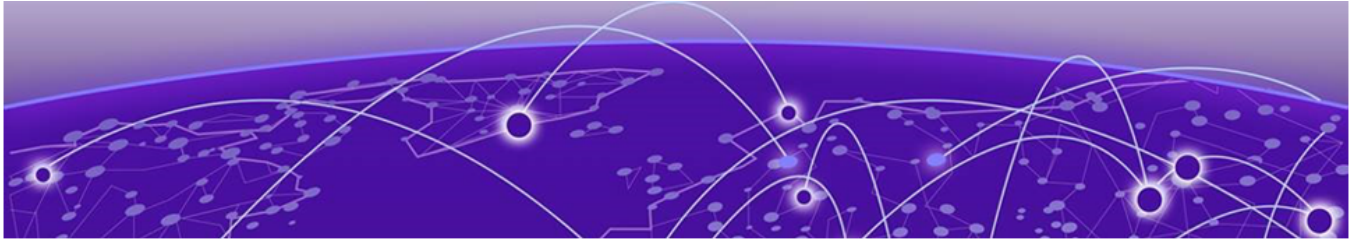
A debug command is available for unit testing purposes that checks health every minute. This is intended for validation and troubleshooting, not normal operation:

```
debug configure switch non-volatile-storage health-check-reporting test
```

To return to normal reporting after using the debug mode:

```
configure switch non-volatile-storage health-check-reporting normal
```

Note
The test mode generates significantly more log entries. Use it only when specifically troubleshooting storage issues with guidance from Technical Support.



STP

Spanning Tree Protocol Overview on page 1787
Span Tree Domains on page 1801
STP Configurations on page 1809
Per VLAN Spanning Tree on page 1816
Rapid Spanning Tree Protocol on page 1817
Multiple Spanning Tree Protocol on page 1829
STP and Network Login on page 1841
STP and ONEPolicy on page 1842
STP and MLAG on page 1842
STP Rules and Restrictions on page 1844
Configuring STP on the Switch on page 1845
Displaying STP Settings on page 1847
STP Configuration Examples on page 1848

Using the *STP (Spanning Tree Protocol)* functionality of the switch makes your network more fault tolerant. This chapter explains more about STP and the STP features supported by ExtremeXOS.

Note
STP is a part of the 802.1D bridge specification defined by the IEEE Computer Society. To explain STP in terms used by the IEEE 802.1D specification, the switch will be referred to as a *bridge*.

ExtremeXOS version 12.0 and later supports the new edition of the IEEE 802.1D standard (known as IEEE 802.1D-2004) for STP, which incorporates enhancements from the IEEE 802.1t-2001, IEEE 802.1W, and IEEE 802.1y standards. The IEEE 802.1D-2004 standard is backward compatible with the IEEE 802.1D-1998 standard. For more information, see Compatibility Between IEEE 802.1D-1998 and IEEE 802.1D-2004 STP Bridges on page 1788.

Spanning Tree Protocol Overview

STP is a bridge-based mechanism for providing fault tolerance on networks.

STP allows you to implement parallel paths for network traffic and to ensure that redundant paths are:

- Disabled when the main paths are operational.

- Enabled if the main path fails.

Note
STP and *ESRP (Extreme Standby Router Protocol)* cannot be configured on the same *VLAN (Virtual LAN)* simultaneously.

Note
If you are transitioning from EOS to ExtremeXOS, please note that ExtremeXOS blocks on a more granular (VLAN) level, instead of at the port level as EOS does.

Compatibility Between IEEE 802.1D-1998 and IEEE 802.1D-2004 STP Bridges

The IEEE 802.1D-2004 compliant bridges interoperate with the IEEE 802.1D-1998 compliant bridges.

To ensure seamless operation of your *STP* network, read this section before you configure STP on any Extreme Networks device running ExtremeXOS 11.6 or later.

Differences in behavior between the two standards include the:

- Default port path cost
- Bridge priority
- Port priority
- Edge port behavior

This section describes the bridge behavior differences in more detail.

Default Port Path Cost

The 802.1D-2004 standard modified the default port path cost value to allow for higher link speeds.

A higher link speed can create a situation whereby an 802.1D-1998 compliant bridge could become the more favorable transit path.

For example, in Figure 202, bridge A is the root bridge running the new 802.1D-2004 standard, bridges B and C are running the old 802.1D-1998 standard, and bridges D, E, and F are running the new 802.1D-2004 standard. In addition, all ports are 100 Mbps links. The ports on bridges B and C have a default path cost of 19, and the ports on bridge A, D, E, and F have a default path cost of 200,000.

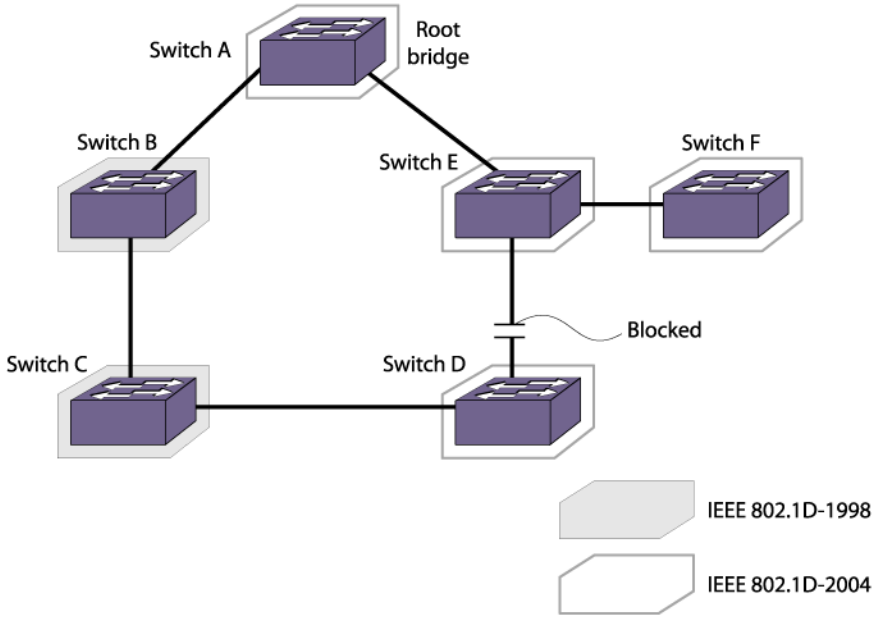


Figure 202: 802.1D-1998 and 802.1D-2004 Mixed Bridge Topology

If you use the default port path costs, bridge D blocks its port to bridge E, and all traffic between bridges D and E must traverse all of bridges in the network. Bridge D blocks its port to bridge E because the path cost to the root bridge is less by going across bridges B and C (with a combined root cost of 38) compared with going across bridge E (with a root cost of 200,000). In fact, if there were 100 bridges between bridges B, C, and D running the old 802.1D-1998 standard with the default port path costs, bridge D would still use that path because the path cost is still higher going across bridge E.

As a workaround and to prevent this situation, configure the port path cost to make links with the same speed use the same path host value. In the example described above, configure the port path cost for the 802.1D-2004 compliant bridges (bridges A, D, E, and F) to 19.

Note

You cannot configure the port path cost on bridges B and C to 200,000 because the path cost range setting for 802.1D-1998 compliant bridges is 1 to 65,535.

To configure the port path cost, use the following command:

```
configure stpd stpd_name ports cost [auto | cost] port_list
```

Bridge Priority

By configuring the *STPD (Spanning Tree Domain)* bridge priority, you make the bridge more or less likely to become the root bridge.

Unlike the 802.1D-1998 standard, the 802.1D-2004 standard restricts the bridge priority to a 16-bit number that must be a multiple of 4,096. The new priority range is 0 to

61,440 and is subject to the multiple of 4,096 restriction. The old priority range was 0 to 65,535 and was not subject to the multiple of 4,096 restriction (except for *MSTP (Multiple Spanning Tree Protocol)* configurations). The default bridge priority remains the same at 32,768.

If you have an ExtremeXOS 11.5 or earlier configuration that contains an *STP* or RSTP bridge priority that is not a multiple of 4,096, the switch rejects the entry and the bridge priority returns to the default value while loading the structure. The MSTP implementation in ExtremeXOS already uses multiples of 4,096 to determine the bridge priority.

To configure the bridge priority, use the following command:

```
configure stpd stpd_name priority priority
```

For example, to lower the numerical value of the priority (which gives the priority a higher precedence), you subtract 4,096 from the default priority: 32,768 - 4,096 = 28,672. If you modify the priority by a value other than 4,096, the switch automatically changes the priority to the lower priority value. For example, if you configure a priority of 31,000, the switch automatically changes the priority to 28,672.

Spanning Tree Protocol (STP) Bridge Priority Incrementing/Decrementing by One

This feature allows you to configure all *STP* modes (RSTP/*MSTP*) bridge priority in increments of 1 or 4,096. Allowing this level of granularity of the priority allows a large range of priority values for the backup root functionality.

After priority reaches 0, backup root is unable to provide for rapid recovery in a lost root situation. In that case, priority returns to its initial configured (or default) value, and the process starts again.

Loss of root should be a rare event, but this feature setting provides a buffer in situations where network downtime to reset the root is not often available.

Supported Platforms

All ExtremeSwitching platforms.

STP Loss of Root Bridge Example

The backup root bridge becomes the root bridge when something goes wrong with the primary root bridge: root bridge failure, root bridge malfunction, or administrative errors, and the backup root bridge looses connectivity with the root bridge.. If the priority of the root bridge is zero and backup root loses the connectivity to the root bridge, then automatic assignment of the priority value for the backup root becomes the initial configured value.

In the following example, Switch1 is the root bridge with priority 32,768. Switch2, Switch3, and Switch4 are configured with priority as 49,152, 36,864, and 40,960, respectively. When the physical link between Switch1 and Switch2 goes down, then Switch2 becomes the root bridge by changing its configured bridge priority, which will be less than Switch1's 32,768 (operational priority); in this case its priority will be 28,672

(32,768 - 4,096). When the priority value reaches 0, then automatic assignment of the priority value returns to the configured value.

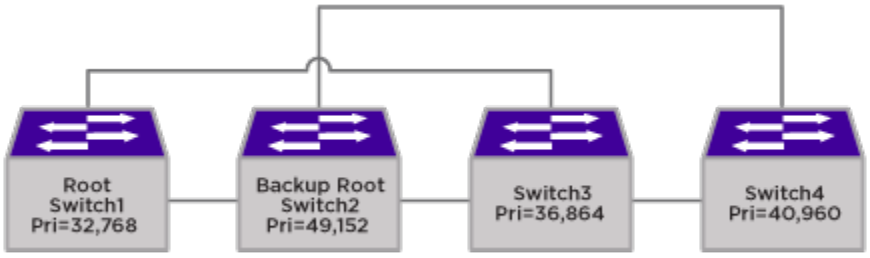


Figure 203: STP Root Bridge Loss Example

Configuring STP Bridge Priority Incrementing/Decrementing by One

To set *STP* bridge priority values, use the following command with the priority-mode set to dot1d:

```
configure stpd stpd_name priority-mode [dot1d | dot1t]
```

Port Priority

The port priority value is always paired with the port number to make up the 16-bit port identifier, which is used in various *STP* operations and the STP state machines.

Unlike the 802.1D-1998 standard, the 802.1D-2004 standard uses only the four most significant bits for the port priority and it must be a multiple of 16. The new priority range available is 0 to 240 and is subject to the multiple of 16 restriction. The 802.1D-1998 standard uses the eight most significant bits for the port priority. The old priority range was 0 to 31 and was not subject to the multiple of 16 restriction.

To preserve backward compatibility and to use ExtremeXOS 11.5 or earlier configurations, the existing `configure stpd ports priority` command is available. If you have an ExtremeXOS 11.5 or earlier configuration, the switch interprets the port priority based on the 802.1D-1998 standard. If the switch reads a value that is not supported in ExtremeXOS 11.6 or later, the switch rejects the entry.

When you save the port priority value, the switch saves it as the command `configure stpd ports port-priority` with the corresponding change in value.

For example, if the switch reads the `configure stpd ports priority 16` command from an ExtremeXOS 11.5 or earlier configuration, (which is equivalent to the command `configure stpd ports priority 8` entered through CLI), the switch saves the value as `configure stpd ports port-priority 128`.

Edge Port Behavior

In ExtremeXOS 11.5 or earlier, Extreme Networks had two edge port implementations: edge port and edge port with safeguard.

The 802.1D-2004 standard has a bridge detection state machine, which introduced a third implementation of edge port behavior. The following list describes the behaviors of the different edge port implementations:

- Edge port (ExtremeXOS 11.5 and earlier):
  - The port does not send bridge protocol data units (BPDUs).
  - The port does not run a state machine.
  - If BPDUs are received, the port discards the BPDU and enters the blocking state.
  - If subsequent BPDUs are not received, the port remains in the forwarding state.
- Edge port with safeguard configured (ExtremeXOS 11.5 and 11.4 only):
  - The port sends BPDUs.
  - When configured for *MSTP*, the port runs a partial state machine.
  - If BPDUs are received, the port enters the blocking state.
  - If subsequent BPDUs are not received, the port attempts to enter the forwarding state.
- Edge port running 802.1D-2004 with safeguard enabled:
  - The port sends BPDUs.
  - The port runs a state machine.
  - If BPDUs are received, the port behaves as a normal RSTP port by entering the forwarding state and participating in RSTP.
  - If subsequent BPDUs are not received, the port attempts to become the edge port again.

Edge port with safeguard prevents accidental or deliberate misconfigurations (loops) by having edge ports enter the blocking state upon receiving a BPDU. The 802.1D-2004 standard implements a bridge detection mechanism that causes an edge port to transition to a non-edge port upon receiving a BPDU; however, if the former edge port does not receive any subsequent BPDUs during a pre-determined interval, the port attempts to become an edge port.

If an 802.1D-2004 compliant safeguard port (edge port) connects to an 802.1D-1998 compliant edge port with safeguard configured, the old safeguard port enters the blocking state. Although the new safeguard port becomes a designated port, the link is not complete (and thus no loop is formed) because one side of the link is blocked.

Restricted Role

In a large metro environment, to prevent external bridges from influencing the spanning tree active topology, the following commands have been introduced for Rapid Spanning Tree Protocol (RSTP) and *MSTP*.

- `configure stpd` *`stpd_name`* **`ports restricted-role enable`** *`port_list`*
  - This command enables restricted role on a specified port in the core network to prevent external bridges from influencing the spanning tree active topology.
  - Restricted role should not be enabled with edge mode.
  - stpd_name—Specifies an *STPD* name on the switch.
  - port_list—Specifies one or more ports or slots and ports.

  - Enabling restricted role causes a port to not be selected as a root port, even if it has the best spanning tree priority vector. Such a port is selected as an alternate port after the root port is selected. The restricted role is disabled by default. If set, it can cause a lack of spanning tree connectivity.
  - A network administrator enables restricted role to prevent external bridges from influencing the spanning tree active topology.
- `configure stpd` *`stpd_name`* **`ports restricted-role disable`** *`port_list`*
  - This command disables restricted role on a specified port in the core network.
  - stpd_name—Specifies an STPD name on the switch.
  - port_list—Specifies one or more ports or slots and ports.
  - Restricted role is disabled by default. If set, it can cause a lack of spanning tree connectivity. A network administrator enables restricted role to prevent external bridges from influencing the spanning tree active topology.

Loop Protect

STP depends on continuous reception of Type 2 BPDUs (RSTP/*MSTP*) based on the port role. The designated port transmits BPDUs and the non-designated port receives BPDUs. If one of the ports in a physical redundant topology no longer receives BPDUs, then STP assumes that the topology is loop free. This leads the blocking port from the alternate or backup port becomes designated and moves to a forwarding State causing a loop.

Loop Protect protects the network from loops. The Loop Protect feature is achieved by ports receiving BPDUs (RSTP/MSTP only) on point-to-point ISLs before their states are allowed to become forwarding. Further, if a BPDU timeout occurs on a port, its state becomes listening until a new BPDU is received. In this way, both upstream and downstream facing ports are protected. When a root or alternate port loses its path to the root bridge, due to message age expiration, it takes on the role of designated port and will not forward traffic until a BPDU is received. When a port is intended to be the designated port in an ISL, it constantly proposes and will not forward until a BPDU is received. It will revert to listening if it stops getting a response. Loop Protect also overrides the port admin setting. This protects against misconfiguration (such as disabling STP on a port) and protocol failure by the connected bridge.

Loop Protect has the capability to

- Control port forwarding state based on reception of agreement BPDUs
- Control port forwarding state based on reception of disputed BPDUs
- Disable a port based on frequency of failure events
- Communicate port non-forwarding status through traps.

Note
Loop Protect Traps are not supported in this ExtremeXOS 15.7.

By default, Loop Protect is disabled on all ports.

Loop Protect Port Modes

Ports work in two Loop Protect operational modes.

- If the port has the partner loop protect as capable then it works in full functional mode.
- If the port has the partner loop protect as incapable then it works limited functional mode.

In full mode, when RSTP/*MSTP* BPDUs is received in point-to-point link and the port is designated, a Loop Protect timer is set to 3 times hello time, when this timer expires then port will be moved to blocking state. Limited mode adds a further requirement that the flags field in the BPDU indicates a root role.

Message age expiration and the expiration of the Loop Protect timer are both events for which Loop Protect generates traps and a debug message. In addition, user can configure Loop Protect to forcefully disable port when one or more events occur. When the configured number of events happens within a given window of time, the port will be forced into disable and held there until you manually unlock it.

The following example shows the loop due to the misconfiguration in *STP*.

Figure 204 shows that Switch 1 is elected as Root. Switch 2 and Switch 3 elect the root port. Switch3's port connected to Switch2 is elected as Alternate port and it is port state is in blocking state.

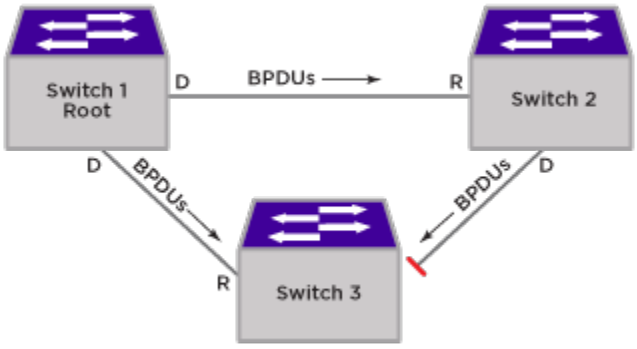


Figure 204: Switch 1 Elected as Root

Figure 205 shows that if the user accidentally disables the STP on Switch 2 port connected to Switch 3, Switch 2 will stop sending the BDPU to Switch 3 since STP is disabled. Switch 3 assumes that neighbor is down and it changes the port to forwarding state which will eventually create a loop.

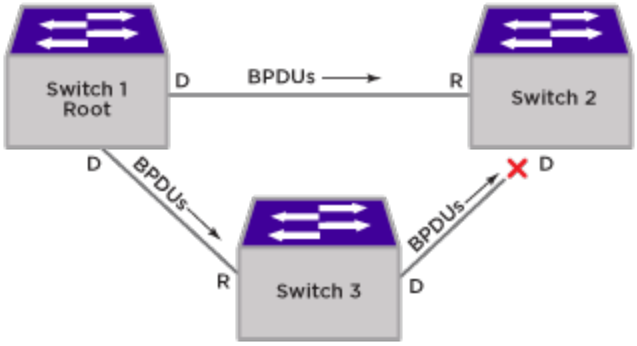


Figure 205: STP Disabled on Switch 2

Figure 206 shows that, with loop protect enabled switch 3 will not go to forwarding state until it receives a BPDU from switch 2 and the port state will be in discarding state.

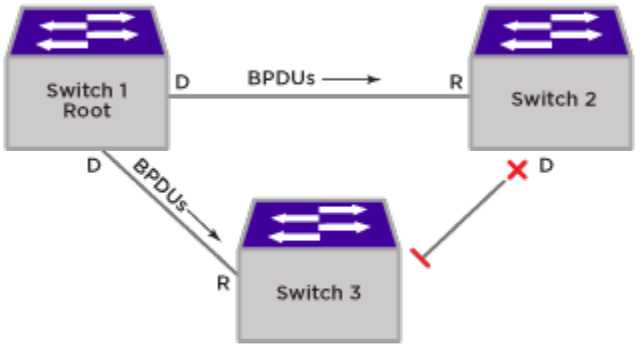


Figure 206: Loop Protect Enabled

When the Loop protect feature is enabled:

- On a Point-to-point Link, BPDU must be received before going to Forwarding state.
- If a BPDU timeout occurs on a port, its state becomes DISCARDING until a BPDU is received.
- When a root or alternate port loses its path to the root bridge due to a message age expiration it takes on the role of designated port. It will not forward traffic until a BPDU is received.
- When a port is intended to be the designated port in a point-to-point link it constantly proposes and will not forward until a BPDU is received, and will revert to discarding if it fails to get a response.

- If the partner is not Loop Protect Capable (Alternate Agreement not supported), designated port will not be allowed to forward unless receiving agreements from a port with root role.
- Legacy Spanning Tree (802.1d) or shared media devices should be connected in a non-redundant fashion to avoid the possibility of looping.

You can enable the port by giving the command `enable port` *`port-list`*.

STP Filter Configuration

This new ExtremeXOS 15.7.1 CLI feature includes processing and forwarding of *STP* BPDU's either by system wide or port based filter installation.

In previous releases this processing and forwarding of STP BPDUs was based only on ports, now this has been changed to system wide (default). Prior to ExtremeXOS 15.7, blocked ports transmission of the BPDU's was restricted, but for the Loop Protect feature when any proposal agreement BPDU is received in the blocked port, then reply for the proposal agreement is to be transmitted. Due to this MAC movement is reported and the packet was not processed. This feature prevents this issue.

You can roll back to the previous implementation (port-based) if any issues are encountered .

Backup Root

When the root bridge of spanning tree instance is lost, its information may be retained in the network until the aging mechanism causes it to be removed. This leads to a delay in convergence on for the new root. This backup root feature is used to get faster convergence when the root bridge connectivity is lost.

Backup root feature enabled bridge port should be connected to Root with point to point link. When backup root bridge lost contact with the root bridge, automatically backup root bridge lowers its bridge priority (an increment below the priority of the lost root). This will cause the backup root bridge to become the new root with information superior to any information held by any bridge in the network. While save and reboot automatically changed priority will not be saved and restored. At the time of reboots it will have its priority restored to the original configured value by administrator.

If the priority of the root bridge is Zero and backup root has lost the connectivity of the root bridge then automatic assignment of the priority value for the backup root will be the initial configured value.

This backup root feature is activated only when there is loss of connectivity with the root bridge. Raising the priority on the root will not cause the Backup root feature to be activated.

Configuration Recommendations:

- Enable the Backup Root feature on both the root and backup root.
- Configure all bridges except the root and backup root with the maximum bridge priority value (61440 with 802.1t).

- Configure the root and backup root to have the next lowest priority (57344 with 802.1t).
- We also recommend having multiple links between the root and backup root. This helps prevent the Backup Root feature from activating due to a simple link failure rather than a bridge failure.
- Care must be taken when this feature is used as it may result in sub-optimal traffic forwarding paths.

  Consider the following topology: 10G link connected from the edge switches to the aggregation Root while the edge switches to the Backup Root switch with 1G links to reduce the cost and acts as a backup link. In this case, when the link between aggregation Root and Backup Root fails, the traffic will now use the 1G links providing reduced performance of the network. The aggregation old root is still in the network which can be used to forward the traffic. This feature has moved the root of the stp domain.

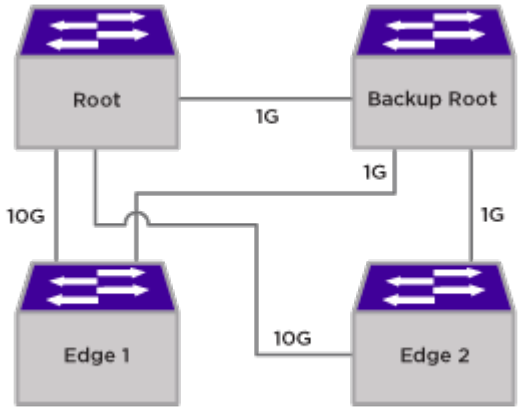


Figure 207: Root and Backup Root Topology

Multisource Detection

Multisource Detection is a feature that prevents network disruption due to excessive topology changes caused by a full duplex port transmitting multiple BPDUs with different source MAC addresses, and different BPDU information. When a port is point-to-point, the received priority information comes from the most recently received BPDU. When a port is non-point-to-point, the received information reflects the best priority information out of all the received BPDUs. Typical scenarios for multisource detection are when a switch is connected to a device which has been improperly configured to forward received BPDUs out on other ports or has been configured to not run the Spanning Tree protocol and treats BPDUs as multicast packets by transmitting them out all other forwarding ports. In these situations, the connected port is acting as a shared media device. The way to detect shared media is the duplex setting. Since the port is full duplex it treats the connection as point-to-point.

When Loop Protect is configured for the port, if multisource detection is triggered, the port will go to the listening state and no longer be part of the active topology. Loop protect does not operate on shared media ports.

Restrict Topology Change Notification

Restricted Topology Change Notification (TCN) is a Spanning Tree Protocol feature that allows/disallows TCN propagation on specified ports. When Restricted TCN is disabled, TCN propagation is allowed. The port propagates received TCNs and topology changes to other ports.

Restricted TCN is disabled by default. When Restricted TCN is enabled, the port does not propagate received TCNs and topology changes to other ports. Enable Restricted TCN to prevent unnecessary address flushing in the core region of the network caused by activation of bridges external to the core network.

A possible reason for not allowing TCN propagation is when bridges are not under the full control of the administrator or because MAC operational state for the attached or downstream LANs transitions frequently, causing disruption throughout the network. Rapid Spanning Tree responds to TCNs by selectively flushing the filter database. Persistent TCNs are disruptive, causing persistent address flushing, which in turn causes increased flooding in the network.

Restricted TCN is a useful tool when it is not possible to remove the source of the TCNs.

BPDU Restrict on Edge Safeguard

BPDU restrict causes a port on which this feature is configured to be disabled as soon as an *STP* BPDU is received on that port, thus allowing you to enforce the STP domain borders and keep the active topology predictable.

Figure 208 shows a BPDU restrict example.

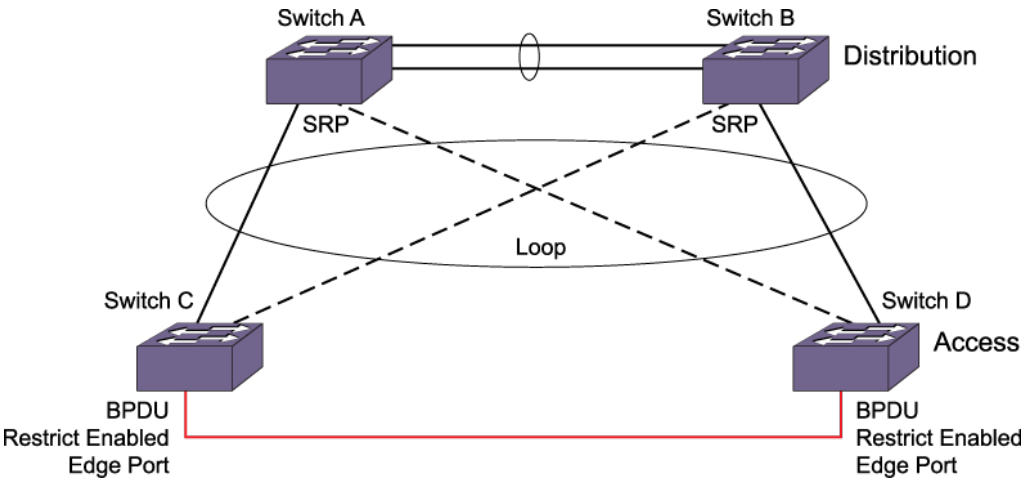


Figure 208: BPDU Restrict

In this figure, loops on the LAN access switches are not prevented since the ports towards the distribution switches are not running STP but Software Redundant Ports

(SRP). Currently, ExtremeXOS software cannot run STP on ports that are configured for SRP. STP on the access switch is unaware of the alternate path and therefore cannot prevent the loop that exists across the switches. Configuring a port as an edge mode port alone cannot prevent the loop between the switches because edge ports never send BPDUs. The edge safeguard feature is not able to prevent the loops because STP does not have the information about the alternate path.

To prevent the loops across the switches, the edge safeguard feature can be configured with the BPDU restrict function. When running in BPDU restrict mode, edge safeguard ports send STP BPDUs at a rate of one every two seconds. The port is disabled as soon as an STP BPDU is received on the BPDU restrict port, thereby preventing the loop. Flexibility is provided with an option to re-enable the port after a user specified time period. If a user enables a port while STP has disabled it, the port is operationally enabled; STP is notified and then stops any recovery timeout that has started.

When an *STPD* is disabled for a BPDU restrict configured port, an STP port in 802.1D operation mode begins forwarding immediately, but in the RSTP or *MSTP* operation modes, the port remains in the disabled state.

BPDU restrict is available on all of the three operational modes of STP: 802.1D, RSTP, and MSTP.

Although edge safeguard is not available in 802.1D operation mode, when you configure BPDU restrict you do so in a similar way, that is, as an extension of edge safeguard; then only BPDU restrict is available on the port and not edge safeguard.

To configure BPDU restrict, use the command:

- `configure {stpd} stpd_name ports edge-safeguard enable port_list {bpdu-restrict} {recovery-timeout {seconds}}`
- BPDU restrict can also be configured by using the following commands:
  - `configure {stpd} stpd_name ports bpdu-restrict [enable | disable] port_list {recovery-timeout {seconds}}`
  - `configure stpd stpd_name ports link-type [[auto | broadcast | point-to-point] port_list | edge port_list {edge-safeguard [enable | disable] {bpdu-restrict} {recovery-timeout seconds}}]`

To include BPDU restrict functionality when configuring link types or edge safeguard, see Configuring Link Types on page 1819 and Configuring Edge Safeguard on page 1820.

The example below shows a BPDU restrict configuration:

```
configure s1 ports edge-safeguard enable 9 bpdu-restrict recovery-timeout 400.
```

The following is sample output from the `show s1 ports` command resulting from the configuration:

```
switch # show s1 ports
Port   Mode   State      Cost  Flags     Priority Port ID Designated Bridge
9      EMISTP FORWARDING 20000 eDee-w-G-- 128      8009    80:00:00:04:96:26:5f:4e
Total Ports: 1
------------------------ Flags: ----------------------------
```

```
1:                e=Enable, d=Disable
2: (Port role)    R=Root, D=Designated, A=Alternate, B=Backup, M=Master
3: (Config type)  b=broadcast, p=point-to-point, e=edge, a=auto
4: (Oper. type)   b=broadcast, p=point-to-point, e=edge
5:                p=proposing, a=agree
6: (partner mode) d = 802.1d, w = 802.1w, m = mstp
7:                i = edgeport inconsistency
8:                S = edgeport safe guard active
s = edgeport safe guard configured but inactive
8:                G = edgeport safe guard bpdu restrict active in 802.1w and mstp
g = edgeport safe guard bpdu restrict active in 802.1d
9:                B = Boundary, I = Internal
10:               r = Restricted Role
switch # show configuration stp
#
# Module stp configuration.
#
configure mstp region 000496265f4e
configure stpd s0 delete vlan default ports all
disable stpd s0 auto-bind vlan default
create stpd s1
configure stpd s1 mode dot1w
enable stpd s0 auto-bind vlan Default
configure stpd s1 add vlan v1 ports 9 emistp
configure stpd s1 ports mode emistp 9
configure stpd s1 ports cost auto 9
configure stpd s1 ports port-priority 128 9
configure stpd s1 ports link-type edge 9
configure stpd s1 ports edge-safeguard enable 9 recovery-timeout 400
configure stpd s1 ports bpdu-restrict enable 9 recovery-timeout 400
enable stpd s1 ports 9
configure stpd s1 tag 10
enable stpd s1
```

The following is sample output for STP operation mode dot1d from the `show configuration stp` command:

```
switch # show configuration stp
#
# Module stp configuration.
#
configure mstp region region2
configure stpd s0 delete vlan default ports all
disable stpd s0 auto-bind vlan default
create stpd s1
enable stpd s0 auto-bind vlan Default
configure stpd s1 add vlan v1 ports 9 emistp
configure stpd s1 ports mode emistp 9
configure stpd s1 ports cost auto 9
configure stpd s1 ports priority 16 9
configure stpd s1 ports link-type edge 9
configure stpd s1 ports edge-safeguard enable 9 recovery-timeout 400
configure stpd s1 ports bpdu-restrict enable 9 recovery-timeout 400
enable stpd s1 ports 9
configure stpd s1 tag 10
enable stpd s1
```

Disable Forwarding of Spanning Tree Protocol (STP) Bridge Protocol Data Units (BDPUs)

You can disable forwarding of *STP* Bridge Protocol Data Units (BPDUs) when STP is disabled on a switch. When STP is disabled globally, BPDUs received on a port are forwarded to other ports that are part of same *VLAN* causing the switch to transparently forward STP/RSTP/*MSTP* BPDUs. This transparent forwarding causes one-to-many peer in the network trouble where STP disabled. To avoid this transparent forwarding of BPDUs, you can opt to discard the BDPUs when there is no STP configuration enabled on the switch.

To disable STP BDPUs, use the command:

```
configure stpd bpdu-forwarding [on | off]
```

Span Tree Domains

The switch can be partitioned into multiple virtual bridges. Each virtual bridge can run an independent Spanning Tree instance. Each Spanning Tree instance is called a *STPD*. Each STPD has its own root bridge and active path. After an STPD is created, one or more *VLAN*s can be assigned to it.

A physical port can belong to multiple STPDs. In addition, a VLAN can span multiple STPDs.

The key points to remember when configuring VLANs and *STP* are:

- Each VLAN forms an independent broadcast domain.
- STP blocks paths to create a loop-free environment.
- Within any given STPD, all VLANs belonging to it use the same spanning tree.
- To create an STPD, use the command:
  `create stpd stpd_name {description stpd-description}`
- To delete an STPD, use the command:
  `delete stpd stpd_name`

  User-created STPD names are not case-sensitive.

For detailed information about configuring STP and various STP parameters on the switch, see Configuring STP on the Switch on page 1845.

Member VLANs

When you add a *VLAN* to an *STPD*, that VLAN becomes a member of the STPD. The two types of member VLANs in an STPD are:

- Carrier
- Protected

carrier VLAN

In STP, carrier VLANs define the scope of the STPD, including the physical and logical ports that belong to the STPD as well as the 802.1Q tags used to transport EMISTP- or PVST+-encapsulated BPDUs. Only one carrier VLAN can exist in any given STPD.

Protected VLAN

Protected *VLAN*s are all other VLANs that are members of the *STPD*.

These VLANs “piggyback” on the carrier VLAN. Protected VLANs do not transmit or receive *STP* BPDUs, but they are affected by STP state changes and inherit the state of the carrier VLAN. Protected VLANs can participate in multiple STPDs, but any particular port in the VLAN can belong to only one STPD. Also known as non-carrier VLANs.

If you configure *MSTP*, all member VLANs in an MSTP region are protected VLANs. These VLANs do not transmit or receive STP BPDUs, but they are affected by STP state changes communicated by the CIST to the MSTP regions. Multiple spanning tree instances (MSTIs) cannot share the same protected VLAN; however, any port in a protected VLAN can belong to multiple MSTIs. For more information about MSTP, see Multiple Spanning Tree Protocol on page 1829.

Specifying the Carrier VLAN

The following example:

- Creates and enables an *STPD* named s8.
- Creates a carrier *VLAN* named v5.
- Assigns VLAN v5 to STPD s8.
- Creates the same tag ID for the VLAN and the STPD (the carrier VLAN’s ID must be identical to the STPD’s ID).

```
create vlan v5
configure vlan v5 tag 100
configure vlan v5 add ports 1:1-1:20 tagged
create stpd s8
configure stpd s8 add vlan v5 ports all emistp
configure stpd s8 tag 100
enable stpd s8
```

Notice how the tag number for the VLAN v5 (100) is identical to the tag for STPD s8. By using identical tags, you have selected the carrier VLAN. The carrier VLAN's ID is now identical to the STPD's ID.

STPD Modes

An *STPD* has three modes of operation:

- 802.1D mode

  Use this mode for backward compatibility with previous *STP* versions and for compatibility with third-party switches using IEEE standard 802.1D. When configured in this mode, all rapid configuration mechanisms are disabled.

- 802.1w mode

  Use this mode for compatibility with Rapid Spanning Tree (RSTP). When configured in this mode, all rapid configuration mechanisms are enabled. The benefit of this mode is available on point-to-point links only and when the peer is likewise

configured in 802.1w mode. If you do not select point-to-point links and the peer is not configured for 802.1w mode, the STPD fails back to 802.1D mode.

You can enable or disable RSTP on a per STPD basis only; you cannot enable RSTP on a per port basis.

For more information about RSTP and RSTP features, see Rapid Spanning Tree Protocol on page 1817.

- *MSTP* mode

  Use this mode for compatibility with MSTP. MSTP is an extension of RSTP and offers the benefit of better scaling with fast convergence. When configured in this mode, all rapid configuration mechanisms are enabled. The benefit of MSTP is available only on point-to-point links and when you configure the peer in MSTP or 802.1w mode. If you do not select point-to-point links and the peer is not configured in 802.1w mode, the STPD fails back to 802.1D mode.

  You must first configure a CIST before configuring any MSTIs in the region. You cannot delete or disable a CIST if any of the MSTIs are active in the system.

  You can create only one MSTP region on the switch, and all switches that participate in the region must have the same regional configurations. You can enable or disable an MSTP on a per STPD basis only; you cannot enable MSTP on a per port basis.

  If configured in MSTP mode, an STPD uses the 802.1D BPDU encapsulation mode by default. To ensure correct operation of your MSTP STPDs, do not configure EMISTP or PVST+ encapsulation mode for MSTP STPDs.

  For more information about MSTP and MSTP features, see Multiple Spanning Tree Protocol on page 1829.

By default:

- The STPD s0 operates in MSTP CIST mode.
- The default device configuration contains a single STPD called s0.
- User-created STPDs operate in dot1d mode.
- The default *VLAN* is a member of STPD s0 with autobind enabled.

To configure the mode of operation of an STPD, use the following command:

```
configure stpd stpd_name mode [dot1d | dot1w | mstp [cist | msti instance]]
```

Note
Starting with ExtremeXOS 22.2, STP operational mode can be changed while VLANs are associated with an STP domain. In MSTP operation mode, mode change is allowed only for CIST domains.

All STP parameters default to the IEEE 802.1D values, as appropriate.

Encapsulation Modes

You can configure ports within an *STPD* to accept specific BPDU encapsulations.

This *STP* port encapsulation is separate from the STP mode of operation. For example, you can configure a port to accept the PVST+ BPDU encapsulation while running in 802.1D mode.

An STP port has three possible encapsulation modes:

- 802.1D mode

  Use this mode for backward compatibility with previous STP versions and for compatibility with third-party switches using IEEE standard 802.1D. BPDUs are sent untagged in 802.1D mode. Because of this, any given physical interface can have only one STPD running in 802.1D mode. The default port mode for user-created STP domains for Dot1d and Dot1w operation mode is Dot1d.

  This encapsulation mode supports the following STPD modes of operation: 802.1D, 802.1w, and *MSTP*.

- Extreme Multiple Instance Spanning Tree Protocol (EMISTP) mode

  EMISTP mode is proprietary to Extreme Networks and is an extension of STP that allows a physical port to belong to multiple STPDs by assigning the port to multiple *VLAN*s. EMISTP adds significant flexibility to STP network design. BPDUs are sent with an 802.1Q tag having an STPD instance Identifier (STPD ID) in the VLAN ID field.

  This encapsulation mode supports the following STPD modes of operation: 802.1D and 802.1w.

- Per VLAN Spanning Tree (PVST+) mode

  This mode implements PVST+ in compatibility with third-party switches running this version of STP. The STPDs running in this mode have a one-to-one relationship with VLANs and send and process packets in PVST+ format.

  This encapsulation mode supports the following STPD modes of operation: 802.1D and 802.1w.

These encapsulation modes are for STP ports, not for physical ports. When a physical port belongs to multiple STPDs, it is associated with multiple STP ports. It is possible for the physical port to run in different modes for different domains to which it belongs.

If configured in MSTP mode, an STPD uses the 802.1D BPDU encapsulation mode by default. To ensure correct operation of your MSTP STPDs, do not configure EMISTP or PVST+ encapsulation mode for MSTP STPDs.

- To configure the BPDU encapsulation mode for one or more STP ports, use the command:
  - `configure stpd` *`stpd_name`* **`ports mode`** [**`dot1d`** | **`emistp`** | **`pvst-plus`**] *`port_list`*

- To configure the default BPDU encapsulation mode on a per STPD basis, use the command:
  - `configure stpd stpd_name default-encapsulation [dot1d | emistp | pvst-plus]`

Instead of accepting the default encapsulation modes of dot1d for the default STPD s0 and for all other STPDs, this command allows you to specify the type of BPDU encapsulation to use for all ports added to the STPD (if not otherwise specified).

STPD Identifier

An StpdID is used to identify each *STP* domain.

When assigning the StpdID when configuring the domain, ensure that the carrier *VLAN* of that *STPD* does not belong to another STPD. Unless all ports are running in 802.1D mode, an STPD with ports running in either EMISTP mode or PVST+ mode must be configured with an StpdID.

An StpdID must be identical to the VLAN ID of the carrier VLAN in that STP domain. For an 802.1D STPD, the VLAN ID can be either a user-defined ID or one automatically assigned by the switch.

Note
If an STPD contains at least one port not in 802.1D mode, you must configure the STPD with an StpdID.

MSTP uses two different methods to identify the STPDs that are part of the MSTP network. An instance ID of 0 identifies the CIST. The switch assigns this ID automatically when you configure the CIST STPD. An *MSTI (Multiple Spanning Tree Instances)* identifier (MSTI ID) identifies each STP domain that is part of an MSTP region. You assign the MSTI ID when configuring the STPD that participates in the MSTP region. In an MSTP region, MSTI IDs only have local significance. You can reuse MSTI IDs across MSTP regions. For more information about MSTP and MSTP features, see Multiple Spanning Tree Protocol on page 1829.

STP States

Each port that belongs to a member *VLAN* participating in *STP* exists in one of the following states:

Blocking

A port in the blocking state does not accept ingress traffic, perform traffic forwarding, or learn MAC source addresses. The port receives STP BPDUs. During STP initialization, the switch always enters the blocking state.

Listening

A port in the listening state does not accept ingress traffic, perform traffic forwarding, or learn MAC source addresses. The port receives STP BPDUs. This is the first transitional state a port enters after being in the blocking state. The bridge listens for BPDUs from neighboring bridge(s) to determine whether the port should or should not be blocked.

Learning

A port in the learning state does not accept ingress traffic or perform traffic forwarding, but it begins to learn MAC source addresses. The port also receives and processes STP BPDUs. This is the second transitional state after listening. From learning, the port will change to either blocking or forwarding.

Forwarding

A port in the forwarding state accepts ingress traffic, learns new MAC source addresses, forwards traffic, and receives and processes STP BPDUs.

Disabled

A port in the disabled state does not participate in STP; however, it will forward traffic and learn new MAC source addresses.

Binding Ports

There are two ways to bind (add) ports to an *STPD*: manually and automatically. By default, ports are manually added to an STPD.

Note
The default *VLAN* and STPD S0 are already on the switch.

Manually Binding Ports

- To manually bind ports, use the commands:
  `configure stpd stpd_name add vlan vlan_name ports [all | port_list] {[dot1d | emistp | pvst-plus]}`

  `configure vlan vlan_name add ports [all | port_list] {tagged {} | untagged} stpd stpd_name {[dot1d | emistp | pvst-plus]}`

  The first command adds all ports or a list of ports within the specified *VLAN* to an *STPD*. For EMISTP and PVST+, the carrier VLAN must already exist on the same set of ports. The second command adds all ports or a list of ports to the specified VLAN and STPD at the same time. If the ports are added to the VLAN but not to the STPD, the ports remain in the VLAN.

  For EMISTP and PVST+, if the specified VLAN is not the carrier VLAN and the specified ports are not bound to the carrier VLAN, the system displays an error message. If you configure *MSTP* on your switch, MSTP does not need carrier VLANs.

  Note
  The carrier VLAN's ID must be identical to the ID of the *STP* domain.

  If you add a protected VLAN or port, that addition inherits the carrier VLAN's encapsulation mode, unless you specify the encapsulation mode when you execute the `configure stpd add vlan` or `configure vlan add ports stpd` commands. If you specify an encapsulation mode (dot1d, emistp, or pvst-plus), the STP port mode is changed to match; otherwise, the STP port inherits either the carrier VLANs encapsulation mode on that port or the STPD's default encapsulation mode.

  For MSTP, you do not need carrier a VLAN. A CIST controls the connectivity of interconnecting MSTP regions and sends BPDUs across the regions to communicate

region status. You must use the dot1d encapsulation mode in an MSTP environment. For more information about MSTP, see the section Multiple Spanning Tree Protocol on page 1829.

- To remove ports, use the command:

  `configure stpd stpd_name delete vlan vlan_name ports [all | port_list]`

  If you manually delete a protected VLAN or port, only that VLAN or port is removed. If you manually delete a carrier VLAN or port, all VLANs on that port (both carrier and protected) are deleted from that STPD.

To learn more about member VLANs, see Member VLANs on page 1801. For more detailed information about these command line interface (CLI) commands, see the Switch Engine v33.6.1 Command References .

Automatically Binding Ports

- To automatically bind ports to an *STPD* when the ports are added to a *VLAN*, use the command:

  `enable stpd stpd_name auto-bind vlan vlan_name`

  The autobind feature is disabled on user-created STPDs. The autobind feature is enabled on the default VLAN that participates in the default STPD S0.

  For EMISTP or PVST+, when you issue this command, any port or list of ports that you add to the carrier VLAN are automatically added to the STPD with autobind enabled. In addition, any port or list of ports that you remove from a carrier VLAN are automatically removed from the STPD. This feature allows the STPD to increase or decrease its span as ports are added to or removed from a carrier VLAN.

  Note
  The carrier VLAN's ID must be identical to the ID of the *STP* domain.

  Enabling autobind on a protected VLAN does not expand the boundary of the STPD.

  If the same set of ports are members of the protected VLAN and the carrier VLAN, protected VLANs are aware of STP state changes. For example, assume you have the following scenario:

  - Carrier VLAN named v1
  - v1 contains ports 3:1-3:2
  - Protected VLAN named v2
  - v2 contains ports 3:1-3:4

  Since v1 contains ports 3:1-3:2, v2 is aware only of the STP changes for ports 3:1 and 3:2, respectively. Ports 3:3 and 3:4 are not part of the STPD, which is why v2 is not aware of any STP changes for those ports.

  In addition, enabling autobind on a protected VLAN causes ports to be automatically added or removed as the carrier VLAN changes.

  For *MSTP*, when you issue this command, any port or list of ports that gets automatically added to an *MSTI* are automatically inherited by the CIST. In addition, any port or list of ports that you remove from an MSTI protected VLAN are

automatically removed from the CIST. For more information, see Automatically Inheriting Ports--MSTP Only on page 1808.

- To remove ports, enter the command:

  ```
  configure stpd stpd_name delete vlan vlan_name ports [all | port_list]
  ```

  If you manually delete a port from the STPD on a VLAN that has been added by autobind, ExtremeXOS records the deletion so that the port does not get automatically added to the STPD after a system restart.

To learn more about the member VLANs, see Member VLANs on page 1801. For more detailed information about these CLI commands, see the Switch Engine v33.6.1 Command References .

Automatically Inheriting Ports--MSTP Only

In an *MSTP* environment, whether you manually or automatically bind a port to an *MSTI* in an MSTP region, the switch automatically binds that port to the CIST.

The CIST handles BPDU processing for itself and all of the MSTIs; therefore, the CIST must inherit ports from the MSTIs in order to transmit and receive BPDUs. You can only delete ports from the CIST if it is no longer a member of an MSTI.

For more information about MSTP, see Multiple Spanning Tree Protocol on page 1829.

Rapid Root Failover

ExtremeXOS supports rapid root failover for faster *STP* failover recovery times in STP 802.1D mode. If the active root port link goes down, ExtremeXOS recalculates STP and elects a new root port. The rapid root failover feature allows the new root port to immediately begin forwarding, skipping the standard listening and learning phases. Rapid root failover occurs only when the link goes down and not when there is any other root port failure, such as missing BPDUs.

The default setting for this feature is disabled.

- To enable rapid root failover, enter the command:

  ```
  enable stpd stpd_name rapid-root-failover
  ```

- To display the configuration, enter the command:

  ```
  show stpd {stpd_name | detail}
  ```

STP and Hitless Failover—SummitStack Only

When you are using redundancy in a SummitStack, one node assumes the role of primary and the other node assumes the role of backup. The primary executes the switch's management functions, and the backup acts in a standby role. Hitless failover transfers switch management control from the primary to the backup and maintains

the state of *STP*. STP supports hitless failover. You do not explicitly configure hitless failover support; rather, if you have two nodes installed, hitless failover is available.

Note
Not all platforms support hitless failover in the same software release. To verify if the software version you are running supports hitless failover, see the following table in Managing the Switch on page 67. For more information about protocol and platform support for hitless failover, see Understanding Hitless Failover Support on page 89.

To support hitless failover, the primary node replicates STP BPDUs to the backup, which allows the nodes to run STP in parallel. Although both primary and backup node receive STP BPDUs, only the primary transmits STP BPDUs to neighboring switches and participates in STP.

Note
Before initiating failover, review the section Synchronizing Nodes on page 1721 to confirm that both primary and backup nodes are running software that supports the `synchronize` command.

To initiate hitless failover on a network that uses STP:

1. Confirm that the nodes are synchronized and have identical software and switch configurations using the command:

   ```
   show switch {detail}
   ```

   The output displays the status of the primary and backup nodes, with the primary node showing MASTER and the backup node showing BACKUP (InSync).

   If the primary and backup nodes are not synchronized and both nodes are running a version of ExtremeXOS that supports synchronization, proceed to 2.

   If the primary and backup nodes are synchronized, proceed to 3 on page 1809.
2. If the primary and backup nodes are not synchronized, use the `synchronize` command to replicate all saved images and configurations from the primary to the backup.

   After you confirm the nodes are synchronized, proceed to 3.
3. If the nodes are synchronized, use the `run failover` (formerly `run msm-failover`) command to initiate failover.

For more detailed information about verifying the status of the primary and backup nodes, and system redundancy, see Understanding System Redundancy on page 84. For more information about hitless failover, see Understanding Hitless Failover Support on page 89.

STP Configurations

When you assign *VLAN*s to an *STPD*, pay careful attention to the *STP* configuration and its effect on the forwarding of VLAN traffic.

This section describes three types of STP configurations:

- Basic STP
- Multiple STPDs on a single port (which uses EMISTP)
- A VLAN that spans multiple STPDs

Basic STP Configuration

This section describes a basic, 802.1D *STP* configuration. Figure 209 on page 1811 illustrates a network that uses *VLAN* tagging for trunk connections.

The following four VLANs have been defined:

- Sales is defined on switch A, switch B, and switch M.
- Personnel is defined on switch A, switch B, and switch M.
- Manufacturing is defined on switch Y, switch Z, and switch M.
- Engineering is defined on switch Y, switch Z, and switch M.
- Marketing is defined on all switches (switch A, switch B, switch Y, switch Z, and switch M).

Two *STPD*s are defined:

- STPD1 contains VLANs Sales and Personnel.
- STPD2 contains VLANs Manufacturing and Engineering.

The carrier and protected VLANs are also defined:

- Sales is the carrier VLAN on STPD1.
- Personnel is a protected VLAN on STPD1.
- Manufacturing is a protected VLAN on STPD2.
- Engineering is the carrier VLAN on STPD2.
- Marketing is a member of both STPD1 and STPD2 and is a protected VLAN.

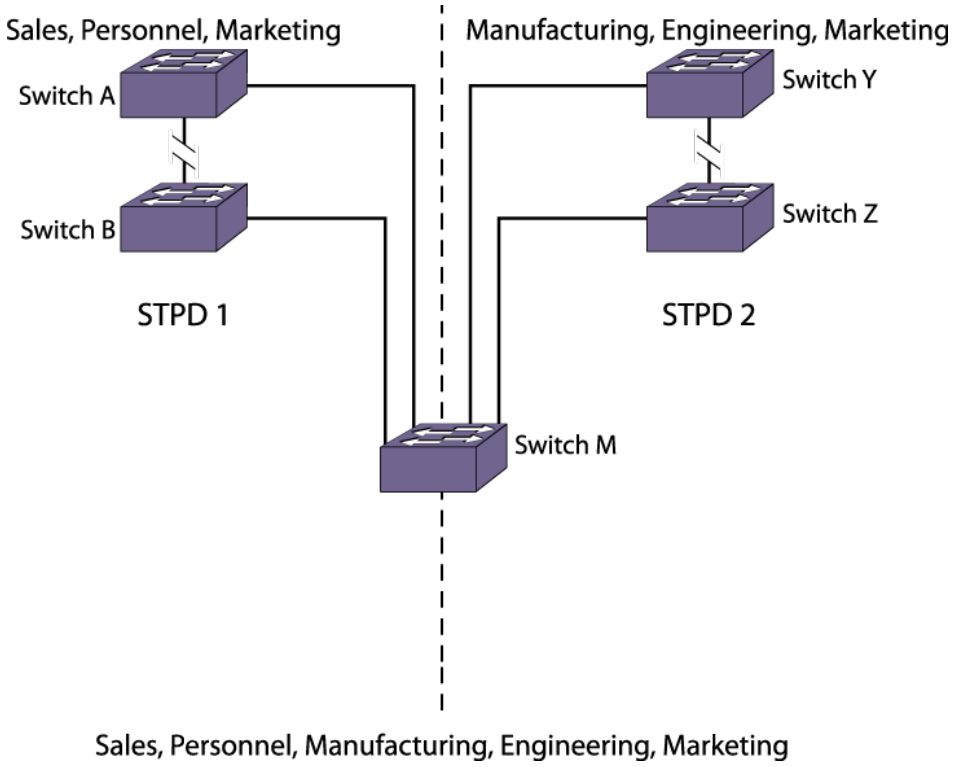


Figure 209: Multiple STPDs

When the switches in this configuration boot-up, STP configures each STPD such that the topology contains no active loops. STP could configure the topology in a number of ways to make it loop-free.

In Figure 209, the connection between switch A and switch B is put into blocking state, and the connection between switch Y and switch Z is put into blocking state. After STP converges, all the VLANs can communicate, and all bridging loops are prevented.

The protected VLAN Marketing, which has been assigned to both STPD1 and STPD2, communicates using all five switches. The topology has no loops, because STP has already blocked the port connection between switch A and switch B and between switch Y and switch Z.

Within a single STPD, you must be extra careful when configuring your VLANs. Figure 210 illustrates a network that has been incorrectly set up using a single STPD so that the STP configuration disables the ability of the switches to forward VLAN traffic.

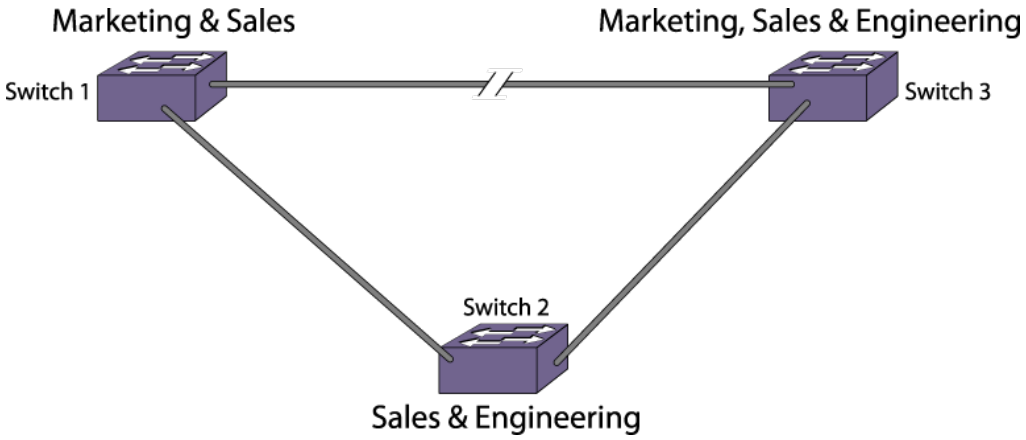


Figure 210: Incorrect Tag-Based STPD Configuration

The tag-based network in the following figure has the following configuration:

- Switch 1 contains VLAN Marketing and VLAN Sales.
- Switch 2 contains VLAN Engineering and VLAN Sales.
- Switch 3 contains VLAN Marketing, VLAN Engineering, and VLAN Sales.
- The tagged trunk connections for three switches form a triangular loop that is not permitted in an STP topology.
- All VLANs in each switch are members of the same STPD.

STP can block traffic between switch 1 and switch 3 by disabling the trunk ports for that connection on each switch.

Switch 2 has no ports assigned to VLAN Marketing. Therefore, if the trunk for VLAN Marketing on switches 1 and 3 is blocked, the traffic for VLAN Marketing will not be able to traverse the switches.

Note
If an STPD contains multiple VLANs, all VLANs should be configured on all ports in that domain, except for ports that connect to hosts (edge ports).

Multiple STPDs on a Port

Traditional 802.1D *STP* has some inherent limitations when addressing networks that have multiple *VLAN*s and multiple *STPD*s.

For example, consider the sample depicted in Figure 211 on page 1813.

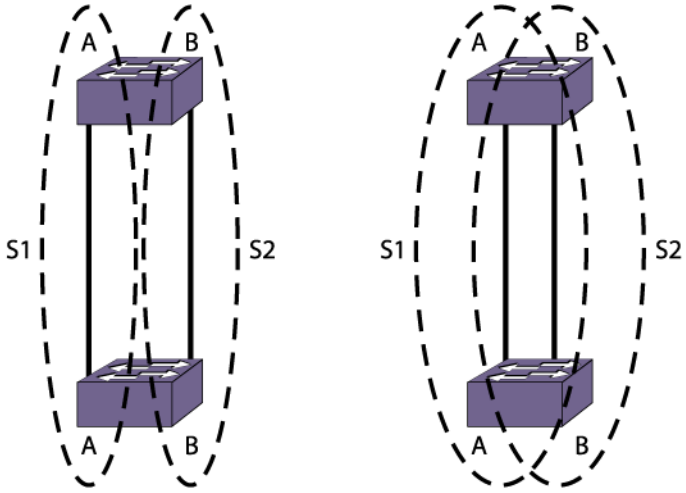


Figure 211: Limitations of Traditional STPD

The two switches are connected by a pair of parallel links. Both switches run two VLANs, A and B. To achieve load-balancing between the two links using the traditional approach, you would have to associate A and B with two different STPDs, called S1 and S2, respectively, and make the left link carry VLAN A traffic while the right link carries VLAN B traffic (or vice versa). If the right link fails, S2 is broken and VLAN B traffic is disrupted.

To optimize the solution, you can use the Extreme Multiple Instance Spanning (EMISTP) mode, which allows a port to belong to multiple STPDs. EMISTP adds significant flexibility to STP network design. Referring to Figure 211, using EMISTP, you can configure all four ports to belong to both VLANs.

Assuming that S1 and S2 still correspond to VLANs A and B respectively, you can fine-tune STP parameters to make the left link active in S1 and blocking in S2, while the right link is active in S2 and blocking in S1. Again, if the right link fails, the left link is elected active by the STP algorithm for S2, without affecting normal switching of data traffic.

Using EMISTP, an STPD becomes more of an abstract concept. The STPD does not necessarily correspond to a physical domain; it is better regarded as a vehicle to carry VLANs that have STP instances. Because VLANs can overlap, so do STPDs. However, even if the different STPDs share the entire topology or part of the redundant topology, the STPDs react to topology change events in an independent fashion.

VLANs Spanning Multiple STPDs

Traditionally, the mapping from *VLAN*s to *STP* instances have been one-to-one or many-to-one.

In both cases, a VLAN is wholly contained in a single instance. In practical deployment there are cases in which a one-to-many mapping is desirable. In a typical large enterprise network, for example, VLANs span multiple sites and/or buildings. Each site represents a redundant looped area. However, between any two sites the topology is usually very simple.

Alternatively, the same VLAN may span multiple large geographical areas (because they belong to the same enterprise) and may traverse a great many nodes.

In this case, it is desirable to have multiple STP domains operating in a single VLAN, one for each looped area.

The justifications include the following:

- The complexity of the STP algorithm increases, and performance drops, with the size and complexity of the network. The 802.1D standard specifies a maximum network diameter of seven hops. By segregating a big VLAN into multiple STPDs, you reduce complexity and enhance performance.
- Local to each site, there may be other smaller VLANs that share the same redundant looped area with the large VLAN. Some STPDs must be created to protect those VLANs. The ability to partition VLANs allows the large VLAN to be "piggybacked" in those STPDs in a site-specific fashion.

Figure 212 has five domains. VLANs green, blue, brown, and yellow are local to each domain. VLAN red spans all of the four domains. Using a VLAN that spans multiple STPDS, you do not have to create a separate domain for VLAN red. Instead, VLAN red is “piggybacked” onto those domains local to other VLANs.

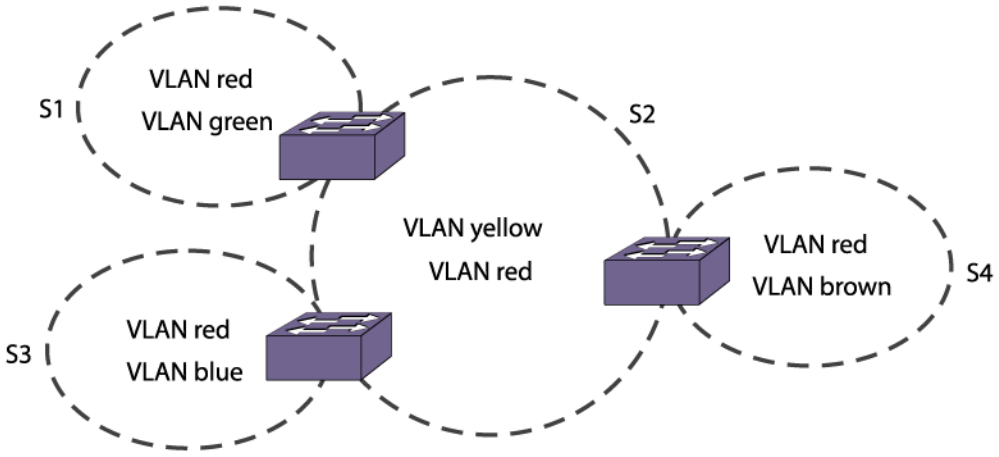


Figure 212: VLANs Spanning Multiple STPDs

In addition, this configuration has these features:

- Each site can be administered by a different organization or department within the enterprise. Having a site-specific STP implementation makes the administration more flexible and convenient.
- Between the sites the connections usually traverse distribution switches in ways that are known beforehand to be “safe” with STP. In other words, the looped areas are already well defined.

EMISTP Deployment Constraints

Although EMISTP greatly enhances *STP* capability, these features must deployed with care.

This section describes configuration issues that, if not followed, could lead to an improper deployment of EMISTP. This section also provides the following restrictive principles to abide by in network design:

- Although a physical port can belong to multiple *STPD*s, any *VLAN* on that port can be in only one domain. Put another way, a VLAN cannot belong to two STPDs on the same physical port.
- Although a VLAN can span multiple domains, any LAN segment in that VLAN must be in the same STPD. VLANs traverse STPDs only inside switches, not across links. On a single switch, however, bridge ports for the same VLAN can be assigned to different STPDs. This scenario is illustrated in Figure 213.

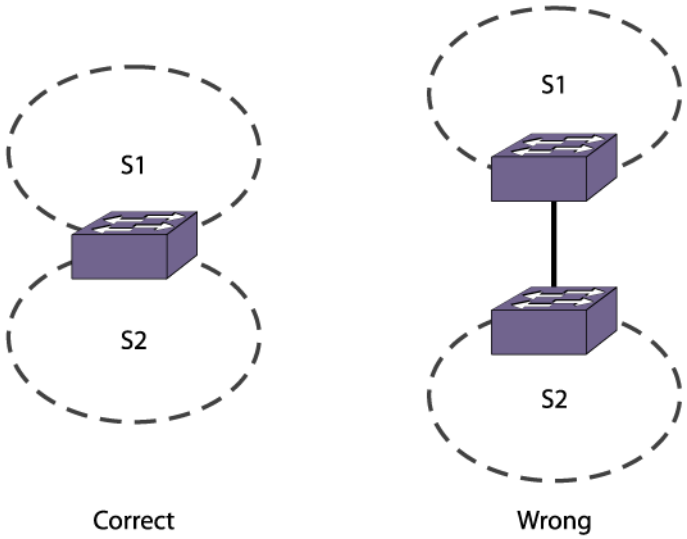


Figure 213: VLANs Traverse Domains Inside Switches

- The VLAN partition feature is deployed under the premise that the overall inter-domain topology for that VLAN is loop-free. Consider the case in Figure 214, VLAN red (the only VLAN in the figure) spans STPDs 1, 2, and 3. Inside each domain, STP produces a loop-free topology. However, VLAN red is still looped, because the three domains form a ring among themselves.

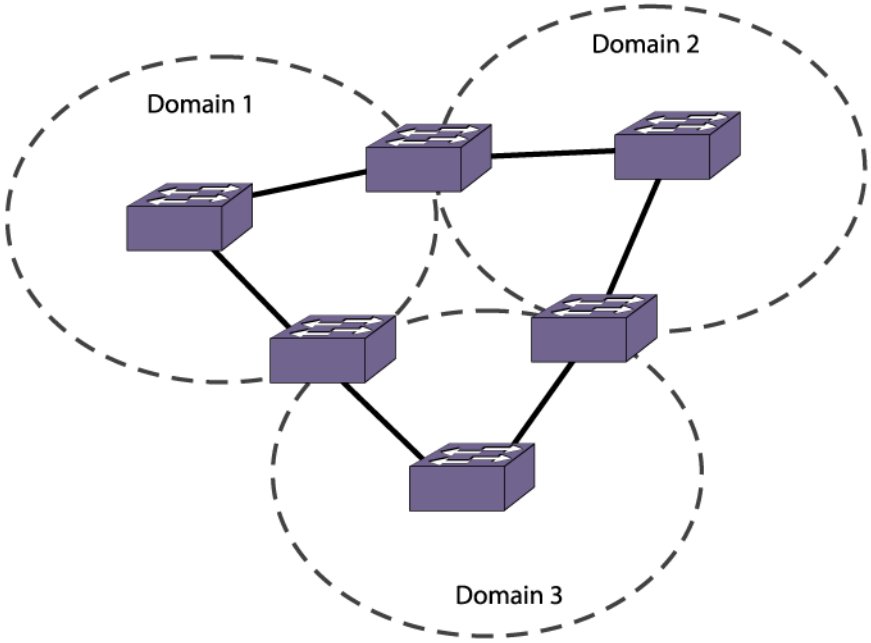


Figure 214: Looped VLAN Topology

- A necessary (but not sufficient) condition for a loop-free inter-domain topology is that every two domains only meet at a single crossing point.

Note
You can use MSTP to overcome the EMISTP constraints described in this section.

Per VLAN Spanning Tree

Switching products that implement Per *VLAN* Spanning Tree (PVST) have been in existence for many years and are widely deployed.

To support *STP* configurations that use PVST, ExtremeXOS has an operational mode called PVST+.

Note
In this document, PVST and PVST+ are used interchangeably. PVST+ is an enhanced version of PVST that is interoperable with 802.1Q STP. The following discussions are in regard to PVST+, if not specifically mentioned.

STPD VLAN Mapping

Each *VLAN* participating in PVST+ must be in a separate *STPD*, and the VLAN number (VLAN ID) must be the same as the STPD identifier (STPD ID).

As a result, PVST+ protected VLANs cannot be partitioned.

This fact does not exclude other non-PVST+ protected VLANs from being grouped into the same STPD. A protected PVST+ VLAN can be joined by multiple non-PVST+ protected VLANs to be in the same STPD.

Note
When PVST+ is used to interoperate with other networking devices, each VLAN participating in PVST+ must be in a separate *STP* domain.

Native VLAN

In PVST+, the native *VLAN* must be peered with the default VLAN on Extreme Networks devices, as both are the only VLANs allowed to send and receive untagged packets on the physical port.

Third-party PVST+ devices send VLAN 1 packets in a special manner. ExtremeXOS does not support PVST+ for VLAN 1. Therefore, when the switch receives a packet for VLAN 1, the packet is dropped.

When a PVST+ instance is disabled, the fact that PVST+ uses a different packet format raises an issue. If the *STPD* also contains ports not in PVST+ mode, the flooded packet has an incompatible format with those ports. The packet is not recognized by the devices connected to those ports.

Rapid Spanning Tree Protocol

The Rapid Spanning Tree Protocol (RSTP), originally in the IEEE 802.1w standard and now part of the IEEE 802.1D-2004 standard, provides an enhanced spanning tree algorithm that improves the convergence speed of bridged networks.

RSTP takes advantage of point-to-point links in the network and actively confirms that a port can safely transition to the forwarding state without relying on any timer configurations. If a network topology change or failure occurs, RSTP rapidly recovers network connectivity by confirming the change locally before propagating that change to other devices across the network. For broadcast links, there is no difference in convergence time between *STP* and RSTP.

RSTP supersedes legacy STP protocols, supports the existing STP parameters and configurations, and allows for seamless interoperability with legacy STP.

RSTP Concepts

Port Roles

RSTP uses information from BPDUs to assign port roles for each LAN segment. Port roles are not user-configurable. Port role assignments are determined based on the following criteria:

- A unique bridge identifier (MAC address) associated with each bridge
- The path cost associated with each bridge port
- A port identifier associated with each bridge port

RSTP assigns one of the following port roles to bridge ports in the network, as described in Table 159.

Table 159: RSTP Port Roles

Port Role	Description
Root	Provides the shortest (lowest) path cost to the root bridge. Each bridge has only one root port; the root bridge does not have a root port. If a bridge has two or more ports with the same path cost, the port with the best port identifier (lowest MAC address) becomes the root port.
Designated	Provides the shortest path connection to the root bridge for the attached LAN segment. To prevent loops in the network, there is only one designated port on each LAN segment. To select the designated port, all bridges that are connected to a particular segment listen to each other's BPDUs and agree on the bridge sending the best BPDU. The corresponding port on that bridge becomes the designated port. If there are two or more ports connected to the LAN, the port with the best port identifier becomes the designated port.
Alternate	Provides an alternate path to the root bridge and the root port.
Backup	Supports the designated port on the same attached LAN segment. Backup ports exist only when the bridge is connected as a self-loop or to a shared-media segment.
Disabled	A port in the disabled state does not participate in RSTP; however, it will forward traffic and learn new MAC source addresses.

When RSTP stabilizes:

- All root ports and designated ports are in the forwarding state.
- All alternate ports and backup ports are in the blocking state.

RSTP makes the distinction between the alternate and backup port roles to describe the rapid transition of the alternate port to the forwarding state if the root port fails.

To prevent a port from becoming an alternate or backup port, use the command:

`configure stpd` *`stpd_name`* **`ports active-role enable`** *`port`* .

To revert to the default that allows a port to be elected to any *STP* port role, use the command:

`configure stpd` *`stpd_name`* **`ports active-role disable`** *`port`*

To view the active-role status, use the command: `show stpd ports`.

Link Types

With *RSTP (Rapid Spanning Tree Protocol)*, you can configure the link type of a port in an *STPD*.

RSTP tries to rapidly move designated point-to-point links into the forwarding state when a network topology change or failure occurs. For rapid convergence to occur, the port must be configured as a point-to-point link.

Table 160 describes the link types.

Table 160: RSTP Link Types

Port Link Type	Description
Auto	Specifies the switch to automatically determine the port link type. An auto link behaves like a point-to-point link if the link is in full-duplex mode or if link aggregation is enabled on the port. Otherwise, the link behaves like a broadcast link used for 802.1w configurations.
Edge	Specifies a port that does not have a bridge attached. An edge port is held in the *STP* forwarding state unless a BPDU is received by the port. In that case, the port behaves as a normal RSTP port. The port is no longer considered an edge port. If the port does not receive subsequent BPDUs during a pre-determined time, the port attempts to become an edge port. ExtremeXOS 11.5 or earlier—An edge port is placed and held in the STP forwarding state unless a BPDU is received by the port. In that case, an edge port enters and remains in the blocking state until it stops receiving BPDUs and the message age timer expires.
Broadcast	Specifies a port attached to a LAN segment with more than two bridges. A port with a broadcast link type cannot participate in rapid reconfiguration using RSTP or *MSTP*. By default, all ports are broadcast links.
Point-to-point	Specifies a port attached to a LAN segment with only two bridges. A port with point-to-point link type can participate in rapid reconfiguration. Used for 802.1w and MSTP configurations.

Configuring Link Types

- To configure the ports in an *STPD*, enter the command:.

  `configure stpd stpd_name ports link-type [[auto | broadcast | point-to-point] port_list | edge port_list {edge-safeguard [enable | disable] {bpdu-restrict} {recovery-timeout seconds}}]`

  Where the following is true:
  - auto—Configures the ports as auto links. If the link is in full-duplex mode or if link aggregation is enabled on the port, an auto link behaves like a point-to-point link.
  - broadcast—Configures the ports as broadcast ports. By default, all ports are broadcast links.
  - point-to-point—Configures the ports for rapid reconfiguration in an RSTP or *MSTP* environment.
  - edge—Configures the ports as edge ports. For information about edge safeguard, see Configuring Edge Safeguard on page 1820.
- To change the existing configuration of a port in an STPD, and return the port to factory defaults, enter the command:

  `unconfigure stpd stpd_name ports link-type port_list`
- To display detailed information about the ports in an STPD, enter the command:

  `show {stpd} stpd_name ports {[detail | port_list {detail}]}`

Configuring Edge Safeguard

Loop prevention and detection on an edge port configured for RSTP is called *edge safeguard*. You can configure edge safeguard on RSTP edge ports to prevent accidental or deliberate misconfigurations (loops) resulting from connecting two edge ports together or by connecting a hub or other non-*STP* switch to an edge port. Edge safeguard also limits the impact of broadcast storms that might occur on edge ports. This advanced loop prevention mechanism improves network resiliency but does not interfere with the rapid convergence of edge ports.

An edge port configured with edge safeguard immediately enters the forwarding state and transmits BPDUs. If a loop is detected, STP blocks the port. By default, an edge port without edge safeguard configured immediately enters the forwarding state but does not transmit BPDUs unless a BPDU is received by that edge port.

You can also configure edge safeguard for loop prevention and detection on an *MSTP* edge port.

- To configure an edge port and enable edge safeguard on that port, use the command:

  `configure stpd` *`stpd_name`* **`ports link-type`** `[[`**`auto`** `|` **`broadcast`** `|` **`point-to-point`**`]` *`port_list`* `|` **`edge`** *`port_list`* `{`**`edge-safeguard`** `[`**`enable`** `|` **`disable`**`]` `{`**`bpdu-restrict`**`}` `{`**`recovery-timeout`** *`seconds`*`}}]`

- If you have already configured a port as an edge port and you want to enable edge safeguard on the port, use the following command:

  `configure {`**`stpd`**`}` *`stpd_name`* **`ports edge-safeguard enable`** *`port_list`* `{`**`bpdu-restrict`**`}` `{`**`recovery-timeout`** `{`*`seconds`*`}}`

- To disable edge safeguard on an edge port, enter the command:

  `configure {`**`stpd`**`}` *`stpd_name`* **`ports edge-safeguard disable`** *`port_list`* `{`**`bpdu-restrict`**`}` `{`**`recovery-timeout`** `{`*`seconds`*`}}`

  `configure stpd` *`stpd_name`* **`ports link-type`** `[[`**`auto`** `|` **`broadcast`** `|` **`point-to-point`**`]` *`port_list`* `|` **`edge`** *`port_list`* `{`**`edge-safeguard`** `[`**`enable`** `|` **`disable`**`]` `{`**`bpdu-restrict`**`}` `{`**`recovery-timeout`** *`seconds`*`}}]`

In ExtremeXOS 11.5 and earlier, ports that connect to non-STP devices are edge ports. Edge ports do not participate in RSTP, and their role is not confirmed. Edge ports immediately enter the forwarding state unless the port receives a BPDU. In that case, edge ports enter the blocking state. The edge port remains in the blocking state until it stops receiving BPDUs and the message age timer expires.

ExtremeXOS 11.6 and later support an enhanced bridge detection method, which is part of the 802.1D-2004 standard. Ports that connect to non-STP devices are still considered edge ports. However, if you have an 802.1D-2004 compliant edge port, the bridge detection mechanism causes the edge port to transition to a non-edge port upon receiving a BPDU. If the former edge port does not receive a subsequent BPDU during a pre-determined interval, the port attempts to become an edge port.

In ExtremeXOS 12.0.3 and 12.1.4 onwards, STP edge safeguard disables a port when a remote loop is detected. ExtremeXOS versions prior to 12.0.3 and 12.1.4 place the port in blocking mode. The change was made because BPDUs are still processed when a port is in a blocking state. A remote loop causes BPDUs to be exponentially duplicated

which caused high CPU utilization on the switch even though the port was transitioned to a blocked state.

Configuring Auto Edge

This feature helps to automatically determine if a port is an edge port or non-edge port. If no BPDU is received for a period of time on auto edge enabled ports, the switch marks those ports as edge ports. If BPDU is received, the switch marks those ports as non-edge ports. By default auto edge is enabled on all ports.

RSTP Timers

For RSTP to rapidly recover network connectivity, RSTP requires timer expiration. RSTP derives many of the timer values from the existing configured *STP* timers to meet its rapid recovery requirements rather than relying on additional timer configurations.

Table 161 describes the user-configurable timers, and Table 162 on page 1821 describes the timers that are derived from other timers and are not user configurable.

Table 161: User-Configurable Timers

Timer	Description
Hello	The root bridge uses the hello timer to send out configuration BPDUs through all of its forwarding ports at a predetermined, regular time interval. The default value is 2 seconds. The range is 1 to 10 seconds.
Forward delay	A port moving from the blocking state to the forwarding state uses the forward delay timer to transition through the listening and learning states. In RSTP, this timer complements the rapid configuration behavior. If none of the rapid rules are in effect, the port uses legacy STP rules to move to the forwarding state. The default is 15 seconds. The range is 4 to 30 seconds.

Table 162: Derived Timers

Timer	Description
TCN	The root port uses the topology change notification (TCN) timer when it detects a change in the network topology. The TCN timer stops when the topology change timer expires or upon receipt of a topology change acknowledgement. The default value is the same as the value for the bridge hello timer.
Topology change	The topology change timer determines the total time it takes the forwarding ports to send configuration BPDUs. The default value for the topology change timer depends upon the mode of the port: 802.1D mode—The sum of the forward delay timer value (default value is 15 seconds; range of 4 to 30 seconds) and the maximum age timer value (default value is 20 seconds; range of 6 to 40 seconds). 802.1w mode—Double the hello timer value (default value is 4 seconds).

Table 162: Derived Timers (continued)

Timer	Description
Message age	A port uses the message age timer to time out receiving BPDUs. When a port receives a superior or equal BPDU, the timer restarts. When the timer expires, the port becomes a designated port and a configuration update occurs. If the bridge operates in 1w mode and receives an inferior BPDU, the timer expires early. The default value is the same as the *STPD* bridge max age parameter.
Hold	A port uses the hold timer to restrict the rate that successive BPDUs can be sent. The default value is the same as the value for the bridge hello timer.
Recent backup	The timer starts when a port leaves the backup role. When this timer is running, the port cannot become a root port. The default value is double the hello time (4 seconds).
Recent root	The timer starts when a port leaves the root port role. When this timer is running, another port cannot become a root port unless the associated port is put into the blocking state. The default value is the same as the forward delay time.

The protocol migration timer is neither user-configurable nor derived; it has a set value of 3 seconds. The timer starts when a port transitions from STP (802.1D) mode to RSTP (802.1w) mode and vice-versa. This timer must expire before further mode transitions can occur.

RSTP Operation

In an RSTP environment, a point-to-point link LAN segment has two bridges.

A switch that considers itself the unique, designated bridge for the attached LAN segment sends a “propose” message to the other bridge to request a confirmation of its role. The other bridge on that LAN segment replies with an “agree” message if it agrees with the proposal. The receiving bridge immediately moves its designated port into the forwarding state.

Before a bridge replies with an “agree” message, it reverts all of its designated ports into the blocking state. This introduces a temporary partition into the network. The bridge then sends another “propose” message on all of its designated ports for further confirmation. Because all of the connections are blocked, the bridge immediately sends an “agree” message to unblock the proposing port without having to wait for further confirmations to come back or without the worry of temporary loops.

Beginning with the root bridge, each bridge in the network engages in the exchange of “propose” and “agree” messages until they reach the edge ports. Edge ports connect to non-*STP* devices and do not participate in RSTP. Their role does not need to be confirmed. If you have an 802.1D-2004 compliant edge port, the bridge detection mechanism causes the edge port to transition to a non-edge port upon receiving a BPDU. If the former edge port does not receive a subsequent BPDU during a pre-determined interval, the port attempts to become an edge port.

RSTP attempts to transition root ports and designated ports to the forwarding state and alternate ports and backup ports to the blocking state as rapidly as possible.

A port transitions to the forwarding state if any of the port:

- Has been in either a root or designated port role long enough that the spanning tree information supporting this role assignment has reached all of the bridges in the network;

  Note
  RSTP is backward-compatible with STP, so if a port does not move to the forwarding state with any of the RSTP rapid transition rules, a forward delay timer starts and STP behavior takes over.

- Is now a root port and no other ports have a recent role assignment that contradicts with its root port role;
- Is a designated port and attaches to another bridge by a point-to-point link and receives an “agree” message from the other bridge port; or
- Is an edge port. An edge port is a port connected to a non-STP device and is in the forwarding state.

The following sections provide more information about RSTP behavior.

Root Port Rapid Behavior

In Figure 215 on page 1824, the diagram on the left displays the initial network topology with a single bridge having the following:

- Two ports are connected to a shared LAN segment.
- One port is the designated port.
- One port is the backup port.

The diagram on the right displays a new bridge that:

- Is connected to the LAN segment.
- Has a superior *STP* bridge priority.
- Becomes the root bridge and sends a *BPDU (Bridge Protocol Data Unit)* to the LAN that is received by both ports on the old bridge.

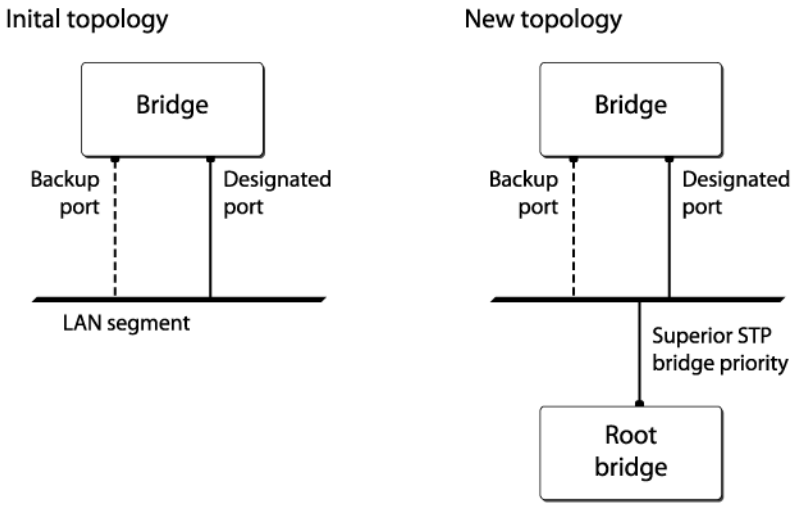


Figure 215: Example of Root Port Rapid Behavior

If the backup port receives the BPDU first, STP processes this packet and temporarily elects this port as the new root port while the designated port's role remains unchanged. If the new root port is immediately put into the forwarding state, there is a loop between these two ports.

To prevent this type of loop from occurring, the recent backup timer starts. The root port transition rule does not allow a new root port to be in the forwarding state until the recent backup timer expires.

Another situation may arise if you have more than one bridge and you lower the port cost for the alternate port, which makes it the new root port. The previous root port is now an alternate port. Depending on your STP implementation, STP may set the new root port to the forwarding state before setting the alternate port to the blocking state. This may cause a loop.

To prevent this type of loop from occurring, the recent root timer starts when the port leaves the root port role. The timer stops if the port enters the blocking state. *RSTP* requires that the recent root timer stop on the previous root port before the new root port can enter the forwarding state.

Designated Port Rapid Behavior

When a port becomes a new designated port, or the *STP* priority changes on an existing designated port, the port becomes an unsynced designated port.

For an unsynced designated port to rapidly move into the forwarding state, the port must propose a confirmation of its role on the attached LAN segment (unless the port is an edge port). Upon receiving an "agree" message, the port immediately enters the forwarding state.

If the receiving bridge does not agree and it has a superior STP priority, the receiving bridge replies with its own BPDU. Otherwise, the receiving bridge keeps silent, and the proposing port enters the forwarding state and starts the forward delay timer.

The link between the new designated port and the LAN segment must be a point-to-point link. If there is a multi-access link, the “propose” message is sent to multiple recipients. If only one of the recipients agrees with the proposal, the port can erroneously enter the forwarding state after receiving a single “agree” message.

Receiving Bridge Behavior

The receiving bridge must decide whether or not to accept a proposal from a port.

Upon receiving a proposal for a root port, the receiving bridge:

- Processes the BPDU and computes the new *STP* topology.
- Synchronizes all of the designated ports if the receiving port is the root port of the new topology.
- Puts all unsynced, designated ports into the blocking state.
- Sends down further “propose” messages.
- Sends back an “agree” message through the root port.

If the receiving bridge receives a proposal for a designated port, the bridge replies with its own BPDU. If the proposal is for an alternate or backup port, the bridge keeps silent.

Propagating Topology Change Information

When a change occurs in the topology of the network, such events are communicated through the network.

In an RSTP environment, only non-edge ports entering the forwarding state cause a topology change. A loss of network connectivity is not considered a topology change; however, a gain in network connectivity must be communicated. When an RSTP bridge detects a topology change, that bridge starts the topology change timer, sets the topology change flag on its BPDUs, floods all of the forwarding ports in the network (including the root ports), and flushes the learned MAC address entries.

Rapid Reconvergence

This section describes the RSTP rapid behavior following a topology change.

In this example, the bridge priorities are assigned based on the order of their alphabetical letters; bridge A has a higher priority than bridge F.

Suppose you have a network, as shown in Figure 216 on page 1826, with six bridges (bridge A through bridge F) where the following is true:

- Bridge A is the root bridge.
- Bridge D contains an alternate port in the blocking state.
- All other ports in the network are in the forwarding state.

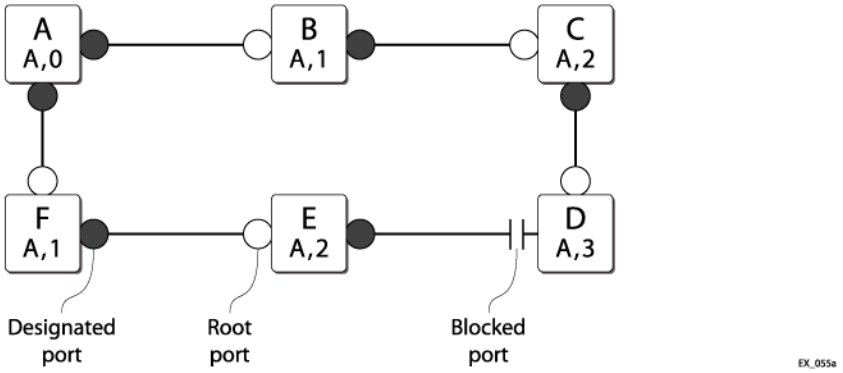


Figure 216: Initial Network Configuration

The network reconverges in the following way:

If the link between bridge A and bridge F goes down, bridge F detects the root port is down. At this point, bridge F:

- Immediately disables that port from the *STP*.
- Performs a configuration update.

As shown in Figure 217, after the configuration update, bridge F:

- Considers itself the new root bridge.
- Sends a BPDU message on its designated port to bridge E.

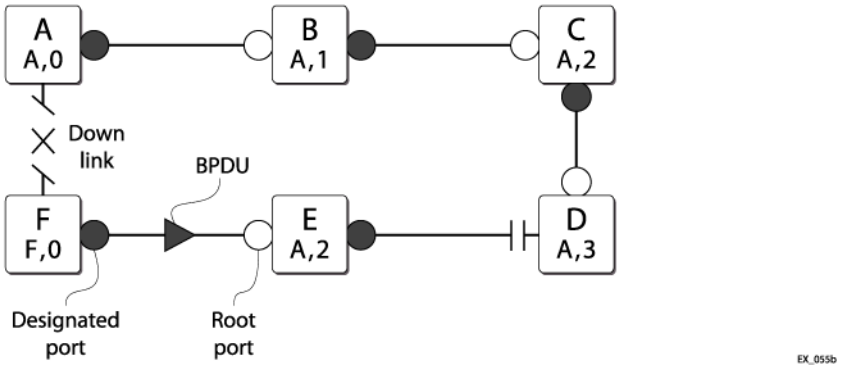


Figure 217: Down Link Detected

- Bridge E believes that bridge A is the root bridge. When bridge E receives the BPDU on its root port from bridge F, bridge E:
  - Determines that it received an inferior BPDU.
  - Immediately begins the max age timer on its root port.
  - Performs a configuration update.

As shown in Figure 218, after the configuration update, bridge E:

- Regards itself as the new root bridge.
- Sends BPDU messages on both of its designated ports to bridges F and D, respectively.

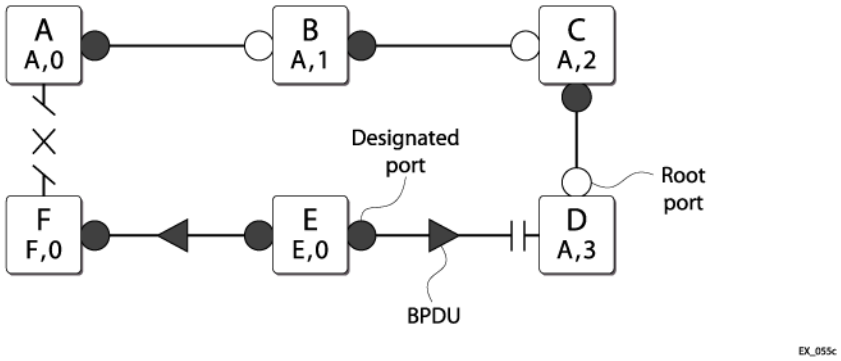


Figure 218: New Root Bridge Selected

As shown in Figure 219, when bridge F receives the superior BPDU and configuration update from bridge E, bridge F:

- Decides that the receiving port is the root port.
- Determines that bridge E is the root bridge.

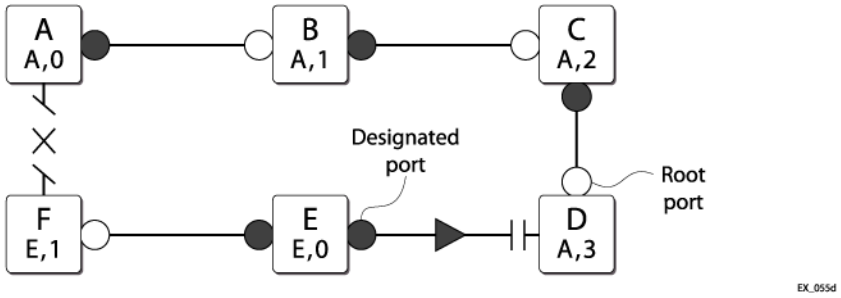


Figure 219: Communicating New Root Bridge Status to Neighbors

Bridge D believes that bridge A is the root bridge. When bridge D receives the BPDU from bridge E on its alternate port, bridge D:

- Immediately begins the max age timer on its alternate port.
- Performs a configuration update.

As shown in Figure 220, after the configuration update, bridge D:

- Moves the alternate port to a designated port.
- Sends a “propose” message to bridge E to solicit confirmation of its designated role and to rapidly move the port into the designated state.

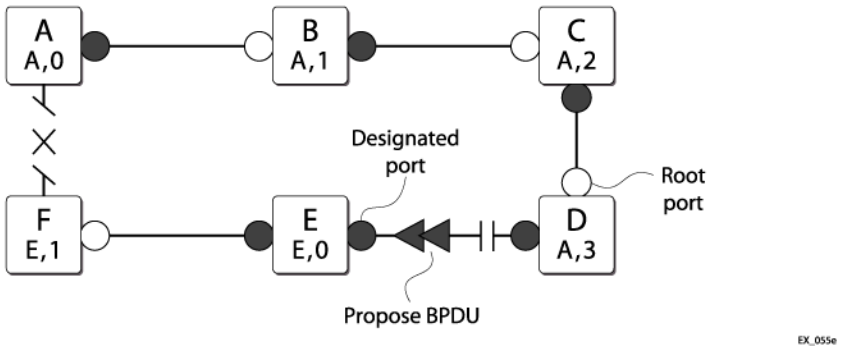


Figure 220: Sending a Propose Message to Confirm a Port Role

Upon receiving the proposal, bridge E (as shown in Figure 221 on page 1828):

- Performs a configuration update.
- Changes its receiving port to a root port.

The existing designated port enters the blocking state.

Bridge E then sends:

- A “propose” message to bridge F.
- An “agree” message from its root port to bridge D.

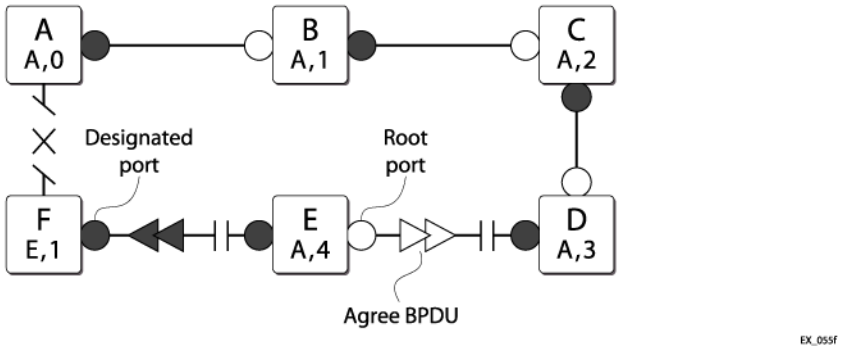


Figure 221: Communicating Port Status to Neighbors

To complete the topology change (as shown in Figure 222):

- Bridge D moves the port that received the “agree” message into the forwarding state.
- Bridge F confirms that its receiving port (the port that received the “propose” message) is the root port, and immediately replies with an “agree” message to bridge E to unblock the proposing port.

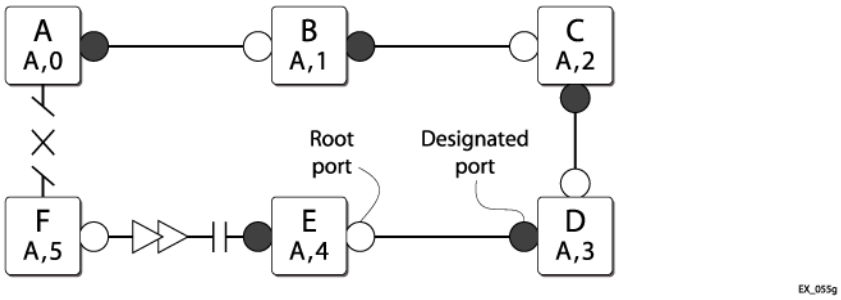


Figure 222: Completing the Topology Change

Figure 223 displays the new topology.

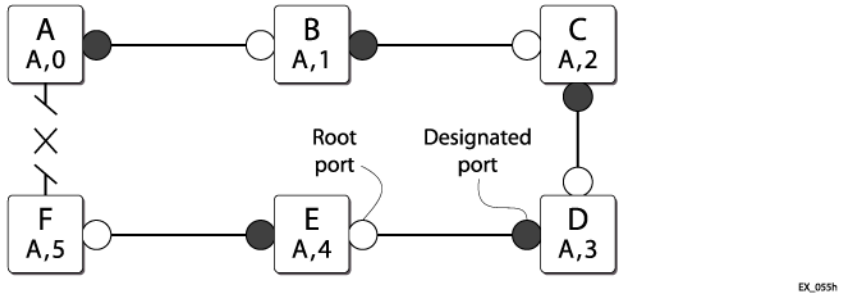


Figure 223: Final Network Configuration

Compatibility With STP (802.1D)

RSTP interoperates with legacy *STP* protocols; however, the rapid convergence benefits are lost when interacting with legacy STP bridges.

Each RSTP bridge contains a port protocol migration state machine to ensure that the ports in the *STPD* operate in the correct, configured mode. The state machine is a protocol entity within each bridge configured to run in 802.1w mode. For example, a compatibility issue occurs if you configure 802.1w mode and the bridge receives an 802.1D BPDU on a port. The receiving port starts the protocol migration timer and remains in 802.1D mode until the bridge stops receiving 802.1D BPDUs. Each time the bridge receives an 802.1D BPDU, the timer restarts. When the port migration timer expires, no more 802.1D BPDUs have been received, and the bridge returns to its configured setting, which is 802.1w mode.

Multiple Spanning Tree Protocol

The *MSTP*, based on IEEE 802.1Q-2003 (formerly known as IEEE 802.1s), allows the bundling of multiple *VLAN*s into one spanning tree topology.

This concept is not new to Extreme Networks. Like MSTP, Extreme Networks proprietary EMISTP implementation can achieve the same capabilities of sharing a virtual network topology among multiple VLANs; however, MSTP overcomes some of the challenges facing EMISTP, including enhanced loop protection mechanisms and new capabilities to achieve better scaling.

MSTP logically divides a Layer 2 network into regions. Each region has a unique identifier and contains multiple spanning tree instances (MSTIs). An *MSTI* is a spanning tree domain that operates within and is bounded by a region. MSTIs control the topology inside the regions. The Common and Internal Spanning Tree (CIST) is a single spanning tree domain that interconnects MSTP regions. The CIST is responsible for creating a loop-free topology by exchanging and propagating BPDUs across regions to form a Common Spanning Tree (CST).

MSTP uses RSTP as its converging algorithm and is interoperable with the legacy *STP* protocols: STP (802.1D) and RSTP (802.1w).

MSTP has three major advantages over 802.1D, 802.1w, and other proprietary implementations:

- To save control path bandwidth and provide improved scalability, MSTP uses regions to localize BPDU traffic. BPDUs containing information about MSTIs contained within an MSTP region do not cross that region's boundary.
- A single BPDU transmitted from a port can contain information for up to 64 STPDs. MSTP BPDU processing utilizes less resources compared to 802.1D or 802.1w where one BPDU corresponds to one *STPD*.
- In a typical network, a group of VLANs usually share the same physical topology. Dedicating a spanning tree per VLAN like PVST+ is CPU intensive and does not scale very well. MSTP makes it possible for a single STPD to handle multiple VLANs.

MSTP Concepts

MSTP Regions

An *MSTP* network consists of either individual MSTP regions connected to the rest of the network with 802.1D and 802.1w bridges or as individual MSTP regions connected to each other.

An MSTP region defines the logical boundary of the network. With MSTP, you can divide a large network into smaller areas similar to an *OSPF (Open Shortest Path First)* area or a *BGP (Border Gateway Protocol)* Autonomous System, which contain a group of switches under a single administration. Each MSTP region has a unique identifier and is bound together by one CIST that spans the entire network. A bridge participates in only one MSTP region at a time.

An MSTP region can hide its internal STPDs and present itself as a virtual 802.1w bridge to other interconnected regions or 802.1w bridges because the port roles are encoded in 802.1w and MSTP BPDUs.

By default, the switch uses the MAC address of the switch to generate an MSTP region. Since each MAC address is unique, every switch is in its own region by default. For multiple switches to be part of an MSTP region, you must configure each switch in the region with the same MSTP region identifiers. See Configuring MSTP Region Identifiers on page 1831 for information.

In Figure 224, all bridges inside MSTP regions 1 and 2 are MSTP bridges; bridges outside of the regions are either 802.1D or 802.1w bridges.

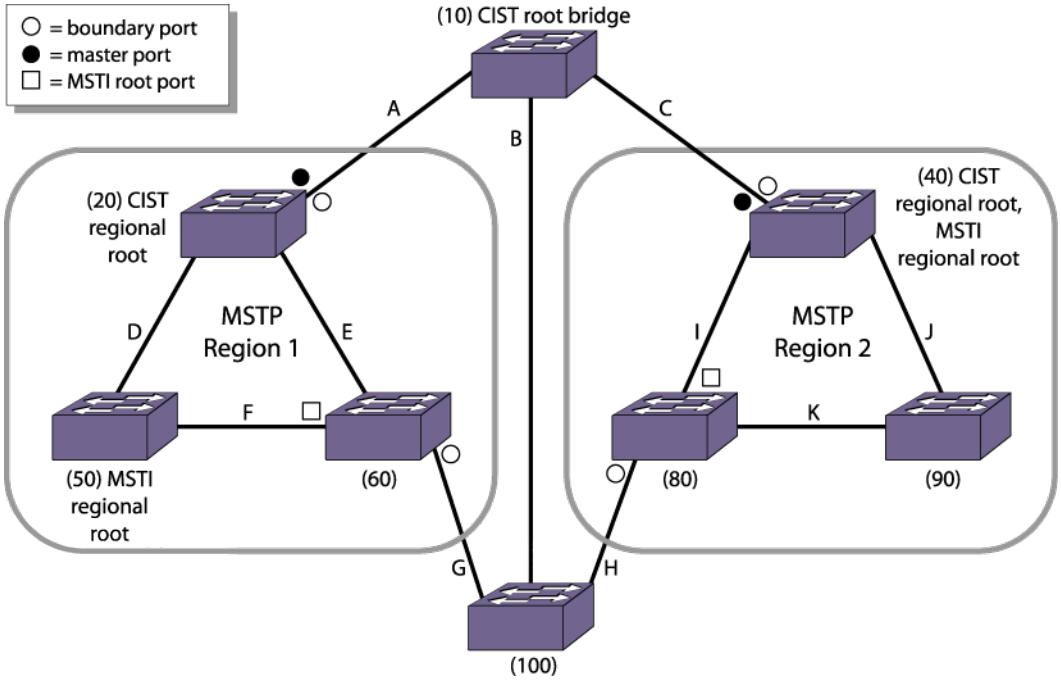


Figure 224: Sample MSTP Topology with Two MSTP Regions

Configuring MSTP Region Identifiers

For multiple switches to be part of an *MSTP* region, you must configure each switch in the region with the same MSTP configuration attributes, also known as MSTP region identifiers. The following list describes the MSTP region identifiers:

- Region Name—This indicates the name of the MSTP region. In the Extreme Networks implementation, the maximum length of the name is 32 characters and can be a combination of alphanumeric characters and underscores (_).
- Format Selector—This indicates a number to identify the format of MSTP BPDUs. The default is 0.
- Revision Level—An unsigned integer encoded within a fixed field of 2 octets that identifies the revision of the current MST configuration. The revision number is not incremented automatically each time that the MST configuration is committed.

The switches inside a region exchange BPDUs that contain information for MSTIs.

The switches connected outside of the region exchange CIST information. By having devices look at the region identifiers, MSTP discovers the logical boundary of a region:

- To configure the MSTP region name, use the command:
  `configure mstp region regionName`

  The maximum length of the region name is 32 characters and can be a combination of alphanumeric characters and underscores (_). You can configure only one MSTP region on the switch at any given time.

  If you have an active MSTP region, we recommend that you disable all active STPDs in the region before renaming the region on all of the participating switches.

- To configure the number used to identify MSTP BPDUs, use the command:

  ```
  configure mstp format format_identifier
  ```

  By default, the value used to identify the MSTP BPDUs is 0. The range is 0 to 255.

  If you have an active MSTP region, we recommend that you disable all active STPDs in the region before modifying the value used to identify MSTP BPDUs on all participating switches.
- To configure the MSTP revision level, use the command:

  ```
  configure mstp revision revision
  ```

  Although this command is available on the CLI, this command is reserved for future use.

Unconfiguring an MSTP Region

Before you unconfigure an *MSTP* region, we recommend that you disable all active STPDs in the region.

To unconfigure the MSTP region on the switch, use the command:

```
unconfigure mstp region
```

After you issue this command, all of the MSTP settings return to their default values. See Configuring MSTP Region Identifiers on page 1831 for information about the default settings.

Common and Internal Spanning Tree

MSTP logically divides a Layer 2 network into regions. The Common and Internal Spanning Tree (CIST) is a single spanning tree domain that interconnects MSTP regions. The CIST is responsible for creating a loop-free topology by exchanging and propagating BPDUs across regions to form a Common Spanning Tree (CST).

In essence, the CIST is similar to having a large spanning tree across the entire network. The CIST has its own root bridge that is common to all MSTP regions, and each MSTP region elects a CIST regional root that connects that region to the CIST, thereby forming a CST.

The switch assigns the CIST an instance ID of 0, which allows the CIST to send BPDUs for itself in addition to all of the MSTIs within an MSTP region. Inside a region, the BPDUs contain CIST records and piggybacked M-records. The CIST records contain information about the CIST, and the M-records contain information about the MSTIs. Boundary ports exchange only CIST record BPDUs.

All MSTP configurations require a CIST domain. You must first configure the CIST domain before configuring any MSTIs. By default, all *MSTI* ports in the region are inherited by the CIST. You cannot delete or disable a CIST if any of the MSTIs are active in the system.

Configuring the CIST

- Configure an *STPD* as the CIST, specifying the mstp cist keywords in the following command:

  ```
  configure stpd stpd_name mode [dot1d | dot1w | mstp [cist | msti instance]]
  ```

  You can enable *MSTP* on a per STPD basis only. By specifying the mstp cist keywords, you can configure the mode of operation for the STPD as MSTP and identify the STPD to be the CIST.

CIST root bridge

In an MSTP environment, the bridge with the lowest bridge ID becomes the CIST root bridge. The bridge ID includes the bridge priority and the MAC address. The CIST root bridge can be either inside or outside an MSTP region. The CIST root bridge is unique for all regions and non-MSTP bridges, regardless of its location.

CIST regional root bridge

Within an MSTP region, the bridge with the lowest path cost to the CIST root bridge is the CIST regional root bridge If the CIST root bridge is inside an MSTP region, that same bridge is the CIST regional root for that region because it has the lowest path cost to the CIST root. If the CIST root bridge is outside an MSTP region, all regions connect to the CIST root through their respective CIST regional roots.

CIST Root Port

The port on the CIST regional root bridge that connects to the CIST root bridge is the *CIST root port* (also known as the master port for MSTIs).

The CIST root port is the master port for all MSTIs in that region, and it is the only port that connects the entire region to the CIST root.

If a bridge is both the CIST root bridge and the CIST regional root bridge, there is no CIST root port on that bridge.

Enabling the CIST

To enable the CIST, use the following command and specify the CIST domain as the *stpd_name*:

```
enable stpd {stpd_name}
```

Multiple Spanning Tree Instances

Multiple spanning tree instances (MSTIs) control the topology inside an *MSTP* region. An *MSTI* is a spanning tree domain that operates within and is bounded by a region; an MSTI does not exchange BPDUs with or send notifications to other regions. You must identify an MSTI on a per region basis. The MSTI ID does not have any significance outside of its region so you can reuse IDs across regions. An MSTI consists of a group of *VLAN*s, which can share the same network topology. Each MSTI has its own root bridge and a tree spanning its bridges and LAN segments.

You must first configure a CIST before configuring any MSTIs in the region. You cannot delete or disable a CIST if any of the MSTIs are active in the system.

You can map multiple VLANs to an MSTI; however, multiple MSTIs cannot share the same VLAN.

Configuring the MSTI and the MSTI ID

MSTP uses the *MSTI* ID, not an Stpd ID, to identify the spanning tree contained within the region. As previously described, the MSTI ID only has significance within its local region, so you can re-use IDs across regions.

To configure the MSTI that is inside an MSTP region and its associated MSTI ID, use the following command and specify the mstp [msti *instance*] parameters:

`configure stpd` *`stpd_name`* **`mode`** [**`dot1d`** | **`dot1w`** | **`mstp`** [**`cist`** | **`msti`** *`instance`*]]

The range of the MSTI instance ID is 1–4094.

MSTP STPDs use 802.1D BPDU encapsulation mode by default. To ensure correct operation of your MSTP STPDs, do not configure EMISTP or PVST+ encapsulation mode for MSTP STPDs. For more information, see Encapsulation Modes on page 1804.

MSTI Regional Root Bridge

Each *MSTI* independently chooses its own root bridge. For example, if two MSTIs are bounded to a region, there is a maximum of two MSTI regional roots and one CIST regional root.

The bridge with the lowest bridge ID becomes the MSTI regional root bridge. The parameters that define the root bridge include the following:

- User-defined bridge priority (by default, the bridge priority is 32,768)
- MAC address

Within an *MSTP* region, the cost from a bridge to the MSTI regional root bridge is known as the MSTI internal path cost. Looking at MSTP region 1 in CIST regional root bridge on page 1833, the bridge with ID 60 has a path cost of F to the MSTI regional root bridge.

The MSTI regional root bridge can be the same as or different from the CIST regional root bridge of that region. You achieve this by assigning different priorities to the *STP* instances configured as the MSTIs and the CIST. For more information about configuring the bridge ID, see the `configure stpd priority` command in the Switch Engine v33.6.1 Command References .

Version 33.4.1 adds support for STP Bridge ID extension in compliance with IEEE 802.1t. This enhancement improves interoperability with third-party vendors by implementing the STP Bridge ID attribute as defined in the IEEE 802.1t standard.

MSTI Root Port

The port on the bridge that has the lowest path cost to the *MSTI* regional root bridge is the MSTI root port.

If a bridge has two or more ports with the same path cost, the port with the best port identifier becomes the root port.

Enabling the MSTI

To enable the *MSTI*, use the following command and specify the MSTI domain as the *stpd_name*:

```
enable stpd {stpd_name}
```

Note
If two switches are configured for the same CIST and MSTI region, in order for them to understand that they are in the same region, both must also belong to the same *VLAN* which is added to the *STP* domain. If they belong to different VLANs, each switch believes that each belongs to a different region. When an *MSTP* BPDU is sent, it carries a VID digest created by VLAN memberships in the CIST domain and the MSTI domain.

Boundary Ports

Boundary ports are bridge ports that are only connected to other *MSTP* regions or 802.1D or 802.1w bridges.

The ports that are not at a region boundary are called internal ports. The boundary ports exchange only CIST BPDUs. A CIST BPDU originated from the CIST root enters a region through the CIST root port and egresses through boundary ports. This behavior simulates a region similar to an 802.1w bridge, which receives BPDUs on its root ports and forwards updated BPDUs on designated ports.

Figure 225 shows an MSTP network that consists of two MSTP regions. Each region has its own CIST regional root and is connected to the CIST root through master ports. The CIST regional roots in each region are the MSTP bridges having the lowest CIST external root path cost. The CIST root is the bridge with the lowest bridge ID and is an 802.1w bridge outside of either MSTP region.

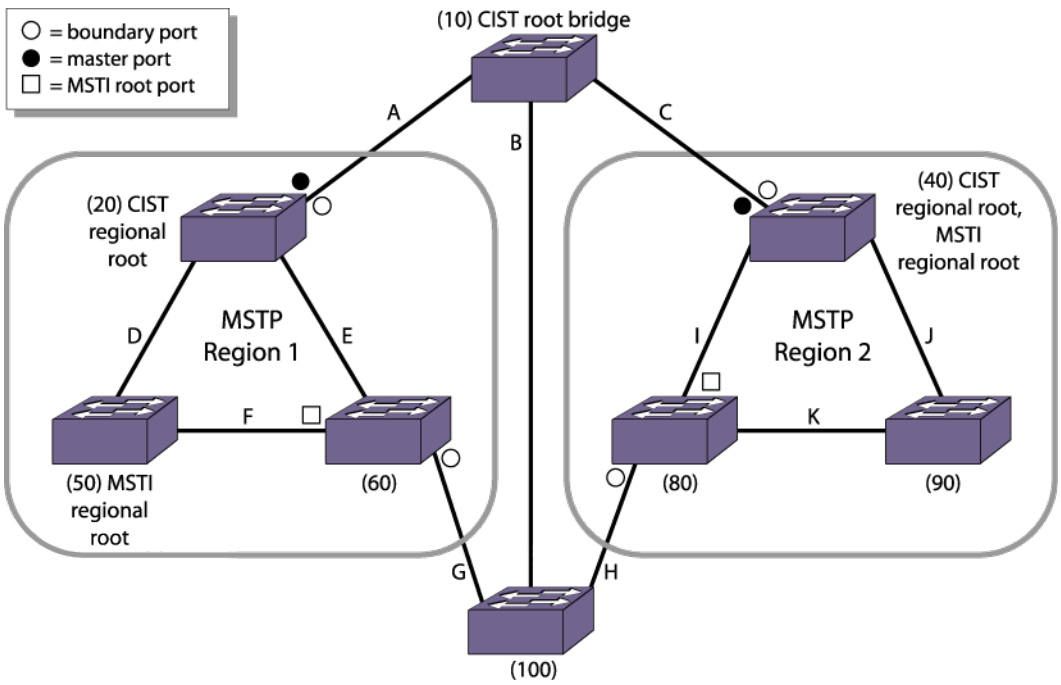


Figure 225: Sample MSTP Topology with Two MSTP Regions

MSTP Region 1 and MSTP Region 2 are connected to the CIST root through directly connected ports, identified as master ports. The bridge with ID 100 connects to the CIST root through Region 1, Region 2, or segment B. For this bridge, either Region 1 or Region 2 can be the designated region or segment B can be the designated segment. The CIST BPDUs egressing from the boundary ports carry the CIST regional root as the designated bridge. This positions the entire MSTP region as one virtual bridge.

The CIST controls the port roles and the state of the boundary ports. A master port is always forwarding for all CIST and *MSTI* VLANs. If the CIST sets a boundary port to the discarding state, the CIST blocks traffic for all VLANs mapped to it and the MSTIs within that region. Each MSTI blocks traffic for their member VLANs and puts their internal ports into the forwarding or blocking state depending on the MSTI port roles.

MSTP Port Roles

MSTP uses the same port roles as RSTP (Root, Designated, Alternate, and Backup).

In addition to these port roles, MSTP introduces a new port role: Master. A Master port is the port that connects an *MSTI* to the CIST root.

MSTP Port States

MSTP uses the same port states as RSTP (Listening, Learning, Forwarding, and Blocking).

In the Extreme Networks MSTP implementation, the listening state is not truly implemented as *FDB (forwarding database)* learning cannot be done when the port is not in the forwarding state. Ports in the blocking state listen but do not accept ingress traffic, perform traffic forwarding, or learn MAC source address; however, the port receives and processes BPDUs.

For more information about all of the *STP* port states, see STP States on page 1805.

MSTP Link Types

MSTP uses the same link types as *STP* and RSTP, respectively.

In an MSTP environment, configure the same link types for the CIST and all MSTIs.

For more information about the link types, see Link Types on page 1818.

MSTP Edge Safeguard

You can configure edge safeguard for loop prevention and detection on an *MSTP* edge port. For more information, see Configuring Edge Safeguard on page 1820.

Note

In MSTP, configuring edge safeguard at CIST will be inherited in all *MSTI*s.
In MSTP, an edge port needs to be added to a CIST before adding it to an MSTI.

MSTP Timers

MSTP uses the same timers as *STP* and RSTP. For more information, see RSTP Timers on page 1821.

MSTP Hop Counts

In an *MSTP* environment, the hop count has the same purpose as the maxage timer for 802.1D and 802.1w environments. The CIST hop count is used within and outside a region. The *MSTI* hop count is used only inside of the region. In addition, if the other end is an 802.1D or 802.1w bridge, the maxage timer is used for interoperability between the protocols.

The BPDUs use hop counts to age out information and to notify neighbors of a topology change.

To configure the hop count.

```
configure stpd stpd_name max-hop-count hopcount
```

By default, the hop count of a BPDU is 20 hops. The range is 6 to 40 hops.

Configuring MSTP on the Switch

To configure and enable *MSTP*:

1. Create the MSTP region using the following command:

   ```
   configure mstp region regionName
   ```

2. Create and configure the CIST, which forms the CST, using the following commands:

   ```
   create stpd stpd_name {description stpd-description}
   configure stpd stpd_name mode mstp cist
   ```

   Note

   You can configure the default *STPD*, S0 as the CIST.

   No *VLAN* can be bound to the CIST and no ports can be added to the CIST. Therefore, the VLAN should be bound to the *MSTI* and the "show MSTI port" command will show the VLAN ports. The ports added to the MSTI are bound automatically to the CIST even though they are not added to it.

3. Enable the CIST using hte command:

   ```
   enable stpd {stpd_name}
   ```

4. Create and configure MSTIs using the commands:

   ```
   create stpd stpd_name {description stpd-description}
   configure stpd stpd_name mode mstp cist instance
   ```

5. Add VLANs to the MSTIs using one of the following commands:

   a. Manually binding ports

      ```
      configure stpd stpd_name add vlan vlan_name ports [all | port_list]
      {[dot1d | emistp | pvst-plus]}
      configure vlan vlan_name add ports [all | port_list] {tagged {} |
      untagged} stpd stpd_name {[dot1d | emistp | pvst-plus]}
      ```

   b. Automatically binding ports to an STPD when ports are added to a member VLAN

      ```
      enable stpd stpd_name auto-bind vlan vlan_name
      ```

6. Enable the MSTIs using the command:.

   ```
   enable stpd {stpd_name}
   ```

For a more detailed configuration example, see MSTP Configuration Example on page 1851.

MSTP Operation

To further illustrate how *MSTP* operates and converges, Figure 226 on page 1839 displays a network with two MSTP regions. Each region contains three MSTP bridges and one *MSTI*. The overall network topology also contains one CIST root bridge (Switch A, which has the lowest bridge ID), one interconnecting 802.1w bridge (Switch D), and 10 full duplex, point-to-point segments. *VLAN* Default spans all of the bridges and segments in the network, VLAN engineering is local to its respective region, and *STPD* S0 is configured as the CIST on all bridges.

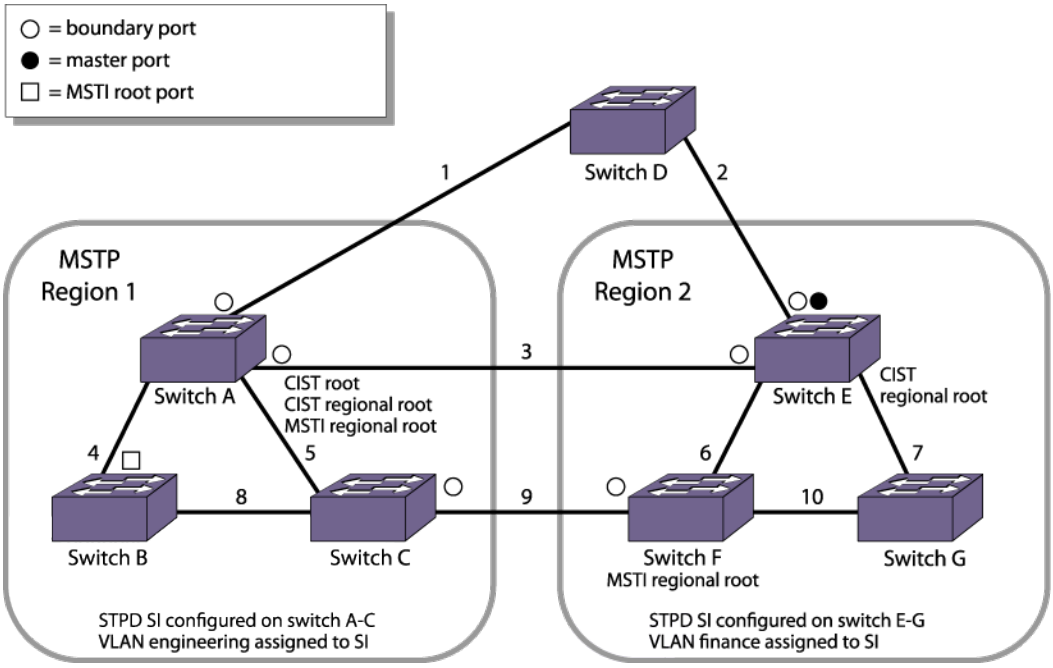


Figure 226: MSTP Topology with the CIST Root Bridge Contained within a Region

MSTP Region 1 consists of the following:

- Three bridges named Switch A, Switch B, and Switch C
- One MSTI STPD named S1 with an MSTI ID of 1
- VLAN Engineering mapped to the MSTI STPD, S1
- Switch A as the CIST root bridge (this is the CIST root bridge for all regions)
- Switch A as the CIST regional root bridge
- Switch A as the MSTI regional root bridge
- Three boundary ports that connect to MSTP Region 2 and other 802.1D or 802.1w bridges

MSTP Region 2 consists of the following:

- Three bridges named Switch E, Switch F, and Switch G
- One MSTI STPD named S1 with an MSTI ID of 1

Note
The MSTI ID does not have any significance outside of its region so you can reuse IDs across regions.

- VLAN finance mapped to the MSTI STPD, S1
- Switch E as the CIST regional root bridge
- Switch F as the MSTI regional root bridge
- One master port that connects to the CIST
- Three boundary ports that connect to MSTP Region 1 and other 802.1D or 802.1w bridges

The following sequence describes how the MSTP topology convergences:

1. Determining the CIST root bridge, MSTP regions, and region boundaries.

   Each bridge believes that it is the root bridge, so each bridge initially sends root bridge BPDUs throughout the network. As bridges receive BPDUs and compare vectors, the bridge with the lowest Bridge ID is elected the CIST root bridge. In our example, Switch A has the lowest Bridge ID and is the CIST root bridge.

   The bridges in the MSTP regions (Switches A, B, C, E, F, and G) advertise their region information along with their bridge vectors.

   Segments 1, 3, and 9 receive BPDUs from other regions and are identified as boundary ports for Region 1. Similarly, segments 2, 3, and 9 are identified as boundary ports for Region 2.
2. Controlling boundary ports.

   The CIST regional root is advertised as the Bridge ID in the BPDUs exiting the region. By sending CIST BPDUs across regional boundaries, the CIST views the MSTP regions as virtual 802.1w bridges. The CIST takes control of the boundary ports and only CIST BPDUs enter or exit a region boundary.

   Each MSTP region has a CIST regional root bridge that communicates to the CIST root bridge. The bridge with the lowest path cost becomes the CIST regional root bridge. The port on the CIST regional root bridge that connects to the CIST root bridge is the CIST root port.

   For Region 1, Switch A has the lowest cost (0 in this example) and becomes the CIST regional root. Since the bridge is both the CIST root bridge and the CIST regional root bridge, there is no CIST root port on the bridge.

   For Region 2, Switch E is the CIST regional root bridge and so a port on that bridge becomes the CIST root port.
3. Identifying MSTI regional roots.

   Each MSTI in a region has an MSTI regional root bridge. MSTI regional roots are selected independently of the CIST root and CIST regional root. The MSTP BPDUs have M-records for each MSTI. Bridges belonging to an MSTI compare vectors in their M-records to elect the MSTI regional root.
4. Converging the CIST.

   The CIST views every region as a virtual bridge and calculates the topology using the 802.1w algorithm. The CIST calculates the topology both inside and outside of a region.
5. Converging MSTIs.

   After the CIST identifies the boundary ports, each MSTI in a domain converge their own trees using 802.1w.

   At this point, all CIST and MSTIs have assigned port roles (Root, Designated, Alternate, and Backup) to their respective spanning trees. All root and designated

ports transition to the forwarding state while the remaining ports remain in the discarding state.

Propagating topology change information is similar to that described for RSTP.

For more information, see Propagating Topology Change Information on page 1825.

For a configuration example, see MSTP Configuration Example on page 1851.

STP and Network Login

STP and network login can be enabled on the same port. This feature can be used to prevent loops while providing redundancy and security on aggregated as well as end switches.

Note
You should be aware that an STP topology change will affect the network login clients. See STP Rules and Restrictions on page 1844 for further information.

Figure 227 shows STP and network login enabled on ports 2 and 3 of Switch 2 and Switch 3 for a typical aggregation scenario.

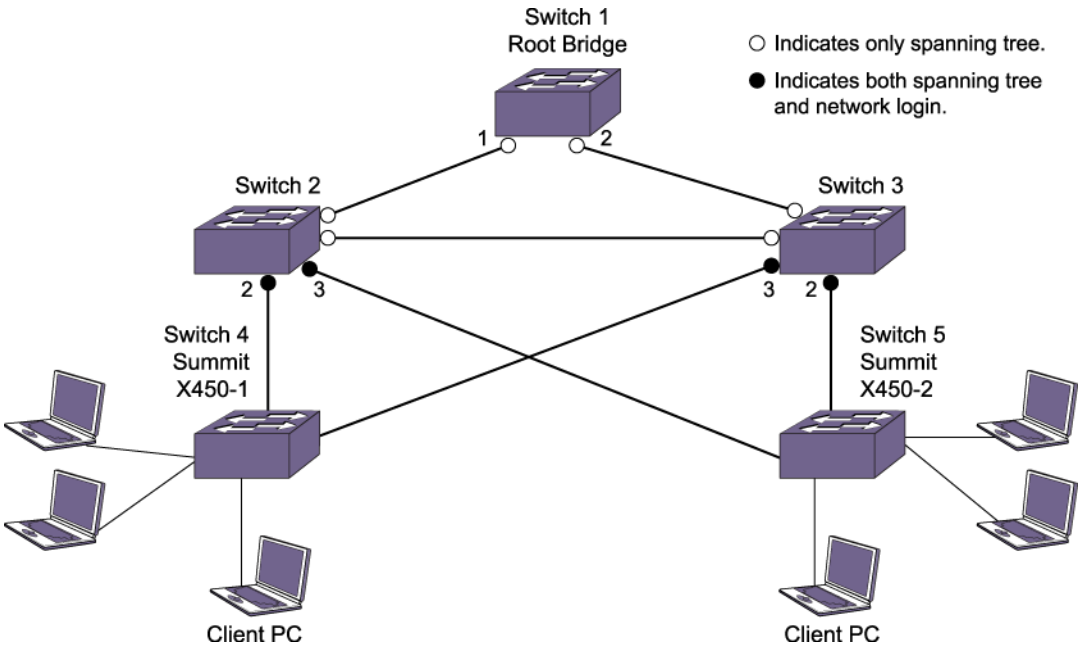


Figure 227: STP and Network Login Enabled

This relieves the administrator from having to configure network login on all the edge ports. All the traffic can be monitored and resiliency is provided at the aggregation side.

The Figure 228 shows a typical scenario for protecting loops and monitoring traffic on the edge side.

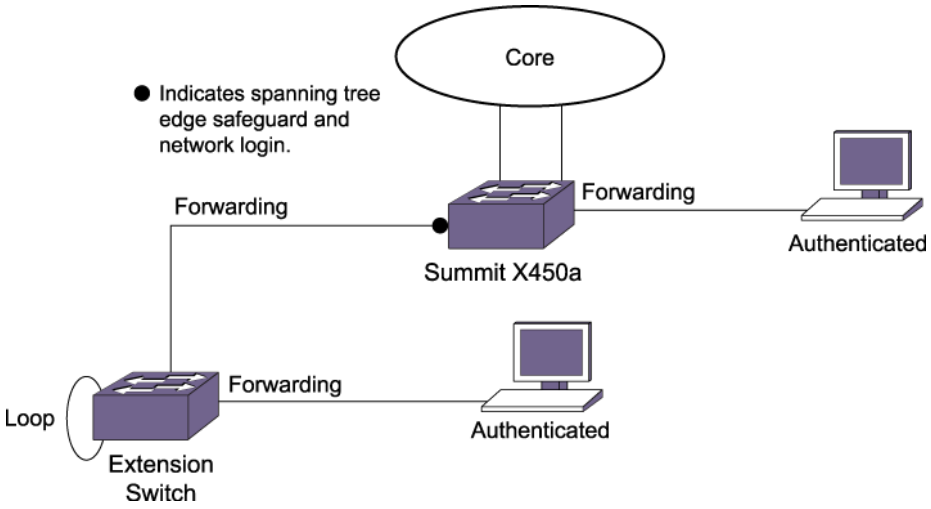


Figure 228: Traffic Monitoring on the Edge Side

In huge networks, it is not easy to control or prevent end users from connecting devices other than workstations to the edge ports. This feature helps prevent the network loops that occur when end users connect a switch or hub to the existing edge port in order to increase the number of end user ports.

STP and ONEPolicy

STP can be configured on ONEPolicy-enabled ports to prevent loop at the edge. In this configuration, it isrecommended to modify the port blocking mechanism to *spanning-tree-group* by running the command **`configure forwarding L2-protocol blocking-type spanning-tree-group`**. To revert to the default blocking mechanism, use the command **`configure forwarding L2-protocol blocking-type ingress-filtering`**.

The blocking type configuration can be verified with the command **`show forwarding configuration`**.

STP and MLAG

You can configure MSTP and RSTP on peers and access switches, which can prevent loops in networks containing MLAG topology. This allows third-party switches to be connected to MLAG topology (as access switches) and an MSTP or RSTP domain can span ExtremeXOS and third-party switches.

In typical MLAG deployments (see Figure 229), connections can exist between access switches, which can cause data loops. By configuring MSTP/RSTP on all the nodes, loops can be effectively prevented.

You should configure the MSTP/RSTP domain in the MLAG peers with a higher priority than the access switches, which blocks the redundant link between access switches. The LAG links in blue form one set of MLAG links and the corresponding ports at MLAG peers are associated with one MLAG index. The LAG links in red form another set of

MLAG links, and the corresponding ports at MLAG peers are associated with another MLAG index.

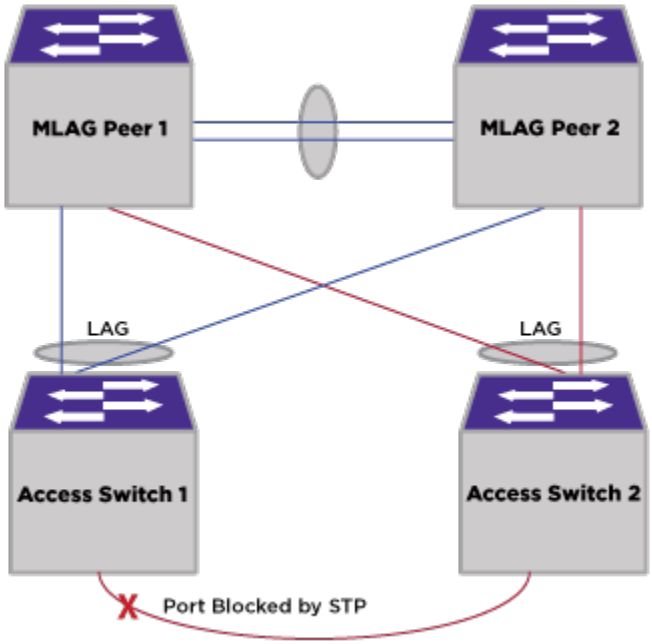


Figure 229: MLAG and MSTP/RSTP

MSTP/RSTP is supported in simple and two-tier MLAG topologies.

For the purpose of synchronization of protocol information, a master is elected for each MLAG index between MLAG peers, based on switch MAC address of peers and MLAG index. By choosing MLAG index with odd or even number, it is possible to change mastership between MLAG peers. Only the master node will transmit BPDUs and process BPDUs. Port roles and states are computed by the master node for the given MLAG index and applied on the local MLAG port. These details are check pointed to the backup node and are applied on its MLAG port.

Recommendations for Configuring MSTP/RSTP with MLAG

You should configure in such a way that one of the MLAG peers become root bridge (see Bridge Priority on page 1789), and also configure MLAG peers with backup root option (see Backup Root on page 1796). The former avoids less powerful access switches from becoming the root bridge; the latter facilitates faster convergence as the other MLAG peer becomes root bridge quicker. Loop protect feature (see Loop Protect on page 1793) should not be enabled on LAG ports (which are part of the MLAG topology) on access switches. This is because a multi-source condition is triggered when one of the two MLAG peers goes down. The loop protect feature prevents the port from going to the forwarding state, if configured. However, enabling this feature on MLAG ports is acceptable.

Since both MLAG peers should act like single switch, the ISC link should not be blocked, regardless of MSTP/RSTP role. Keeping this in mind, the port cost of the ISC port is

chosen internally to be the lowest possible value (see Default Port Path Cost on page 1788). This helps in selecting the ISC link as the active link if there are any parallel links to the ISC. MLAG links do not have special preference over redundant connections that are in parallel to the MLAG link. The standard MST/RSTP algorithm disables both MLAG links if it has higher root path cost than the redundant link. This allows the lower cost link to be chosen as active link. If you configure MSTP over MLAG, both MLAG peers and access switches should be in the same MSTP region. This makes MLAG and ISC ports internal ports.

STP Rules and Restrictions

This section summarizes the rules and restrictions for configuring *STP* are:

- The carrier *VLAN* must span all ports of the *STPD*. (This is not applicable to *MSTP*.)
- The StpdID must be the VLAN ID of the carrier VLAN; the carrier VLAN cannot be partitioned. (This is not applicable to MSTP.)
- A default VLAN cannot be partitioned. If a VLAN traverses multiple STPDs, the VLAN must be tagged.
- An STPD can carry, at most, one VLAN running in PVST+ mode, and its STPD ID must be identical with that VLAN ID. In addition, the PVST+ VLAN cannot be partitioned.
- The default VLAN of a PVST+ port must be identical to the native VLAN on the PVST+ device connected to that port.
- If an STPD contains both PVST+ and non-PVST+ ports, that STPD must be enabled. If that STPD is disabled, the BPDUs are flooded in the format of the incoming STP port, which may be incompatible with those of the connected devices.
- The 802.1D ports must be untagged and the EMISTP/PVST+ ports must be tagged in the carrier VLAN.
- STP and network login operate on the same port as follows:
  - STP (802.1D), RSTP (802.1w), and MSTP (802.1s) support both network login and STP on the same port.
  - At least one VLAN on the intended port should be configured both for STP and network login.
  - STP and Network Login operate together in both Network Login ISP mode and campus mode starting with Extreme 22.2.
  - When STP blocks a port, network login does not process authentication requests. All network traffic, except STP BPDUs, is blocked.
  - When STP places a port in forwarding state, all network traffic is allowed and network login starts processing authentication requests.
- STP cannot be configured on the following ports:
  - *FDB* flush does not occur on STP- and NetLogin-enabled ports during the topology changes.
  - A mirroring target port.
  - A software-controlled redundant port.
- When you are using the older method of enabling STP instead of using EAPSv2 to block the super loop in a shared-port environment, you can continue to do so. In all other scenarios, it is not recommended to use both STP and EAPS on the same port.

- MSTP and 802.1D STPDs cannot share a physical port.
- Only one MSTP region can be configured on a switch.
- In an MSTP environment, a VLAN can belong to one of the MSTIs.
- A VLAN can belong to only one MSTP domain.
- MSTP is not interoperable with PVST+.
- No VLAN can be bound to the CIST.
- Starting with ExtremeXOS 22.2, under the following two conditions, these attributes are preserved with autobind being used to bind the STPD domain to the VLAN: STP enable, Cost, Priority, LinkType, Edge-Sage-Guard, BPDU Restrict, Restrict Role, ActiveRole, LoopProtect, LoopProtect-Partner, AutoEdge, Restricted TCN:
  - With sharing and STP configurations established, disabling sharing no longer removes the STP port configuration for the configured master port and it preserves certain configuration attributes and restores them when the port is recreated.
  - Configured port's STP properties are preserved when the port is moved from one VLAN to another, provided the new port gets associated with same STP domain.
- MLAG is not supported with STP (802.1D).

Configuring STP on the Switch

Note
If you are transitioning from EOS to ExtremeXOS, please note that ExtremeXOS blocks on a more granular (*VLAN*) level, instead of at the port level as EOS does.

To configure basic *STP*:

1. Create one or more *STPD*s using the command:

   ```
   create stpd stpd_name {description stpd-description}
   ```

2. Add one or more VLANs to the STPD using the command:

   ```
   configure stpd stpd_name add vlan vlan_name | vlan_list ports [all | port_list] {[dot1d | emistp | pvst-plus]}
   ```

3. Define the carrier VLAN using the command:.

   ```
   configure stpd stpd_name tag stpd_tag
   ```

   Note
   The carrier VLAN's ID must be identical to the StpdID.

4. Enable STP for one or more STPDs using the command:

   ```
   enable stpd {stpd_name}
   ```

5. After you have created the STPD, you can optionally configure STP parameters for the STPD.

   Note
   You should not configure any STP parameters unless you have considerable knowledge and experience with STP. The default STP parameters are adequate for most networks.

The following parameters can be configured on each STPD:

- Hello time (In an *MSTP* environment, configure this only on the CIST.)
- Forward delay
- Max age (In an MSTP environment, configure this only on the CIST.)
- Max hop count (MSTP only)
- Bridge priority
- Domain description
- StpdID (STP, RSTP, EMISTP, and PVST+ only)
- *MSTI* ID (MSTP only)

The following parameters can be configured on each port:

- Path cost
- Port priority
- Port mode

Note

When using more than one STP domain over the same set of ports, it is recommended to turn off the L2-fast-convergence feature (`configure forwarding L2-protocol fast-convergence off`) as this feature causes unnecessary flooding during topology changes.

Note

The device supports the RFC 1493 Bridge MIB, RSTP-03, and Extreme Networks STP MIB. Parameters of the s0 default STPD support RFC 1493 and RSTP-03. Parameters of any other STPD support the Extreme Networks STP MIB.

If an STPD contains at least one port not in 802.1D (dot1D) mode, the STPD must be configured with an StpdID.

The following section provides more detailed STP configuration examples, including 802.1D, EMISTP, RSTP, and MSTP.

STP FDB Flush Criteria

When there are more than 1000 VLANs and more than 70 ports participating in *STP*, the number of messages exchanged between STP/*FDB*/HAL modules can consume a lot of system memory when trying to flush the FDB during a STP topology change. To help avoid this high consumption, you can set the flush type from the default of vlan-and-port to port-based.

To set the flush type, enter the command:

`configure stpd flush-method [`**`vlan-and-port`**` | `**`port-only`**`]`

Displaying STP Settings

- To display *STPD* settings, use the command:

  ```
  show stpd {stpd_name | detail}
  ```

  To display more detailed information for one or more STPDs, specify the detail option.

  This command displays the following information:
  - STPD name
  - STPD state
  - STPD mode of operation
  - Domain description
  - Rapid Root Failover
  - Tag
  - Ports
  - Active VLANs
  - Bridge priority
  - Bridge ID
  - Designated root
  - STPD configuration information

  If you have *MSTP* configured on the switch, this command displays additional information:
  - MSTP Region
  - Format Identifier
  - Revision Level
  - Common and Internal Spanning Tree (CIST)
  - Total number of MST Instances (*MSTI*)
- To display the state of a port that participates in *STP*, use the command:

  ```
  show {stpd} stpd_name ports {[detail | port_list {detail}]}
  ```

  To display more detailed information for one or more ports in the specified STPD, including participating VLANs, specify the detail option.

  This command displays the following information:
  - STPD port configuration
  - STPD port mode of operation
  - STPD path cost
  - STPD priority
  - STPD state (root bridge, etc.)
  - Port role (root designated, alternate, etc.)
  - STPD port state (forwarding, blocking, etc.)
  - Configured port link type
  - Operational port link type

- Edge port settings (inconsistent behavior, edge safeguard setting)
- MSTP port role (internal or boundary)

If you have MSTP configured and specify the detail option, this command displays additional information:

- MSTP internal path cost
- MSTP timers
- STPD *VLAN* Settings

- If you have a VLAN that spans multiple STPDs, use the `show {vlan} vlan_name stpd` command to display the STP configuration of the ports assigned to that specific VLAN.

  The command displays the following:
  - STPD port configuration
  - STPD port mode of operation
  - STPD path cost
  - STPD priority
  - STPD state (root bridge, etc.)
  - Port role (root designated, alternate, etc.)
  - STPD port state (forwarding, blocking, etc.)
  - Configured port link type
  - Operational port link type

STP Configuration Examples

Note
If you are transitioning from EOS to ExtremeXOS, please note that ExtremeXOS blocks on a more granular (*VLAN*) level, instead of at the port level as EOS does.

Basic 802.1D Configuration Example

The following example:

- Removes ports from the *VLAN* Default that will be added to VLAN Engineering.
- Creates the VLAN Engineering.
- Assigns a VLAN ID to the VLAN Engineering.

  Note
  If you do not explicitly configure the VLAN ID in your 802.1D deployment, use the `show vlan` command to see the internal VLAN ID automatically assigned by the switch.

- Adds ports to the VLAN Engineering.
- Creates an *STPD* named Backbone_st.
- Configures the default encapsulation mode of dot1d for all ports added to STPD Backbone_st.

- Enables autobind to automatically add or remove ports from the STPD.
- Assigns the Engineering VLAN to the STPD.
- Assigns the carrier VLAN.
- Enables *STP*.

Note
To assign the carrier VLAN, the StpdID must be identical to the VLAN ID of the carrier VLAN.

```
configure vlan default delete ports 2:5-2:10
create vlan engineering
configure vlan engineering tag 150
configure vlan engineering add ports 2:5-2:10 untagged
create stpd s1
configure stpd s1 default-encapsulation dot1d
enable stpd s1 auto-bind vlan engineering
configure stpd s1 tag 150
enable stpd s1
```

By default, the port encapsulation mode for user-defined STPDs is dot1d. In this example, you set it to dot1d.

EMISTP Configuration Example

Figure 230 is an example of EMISTP.

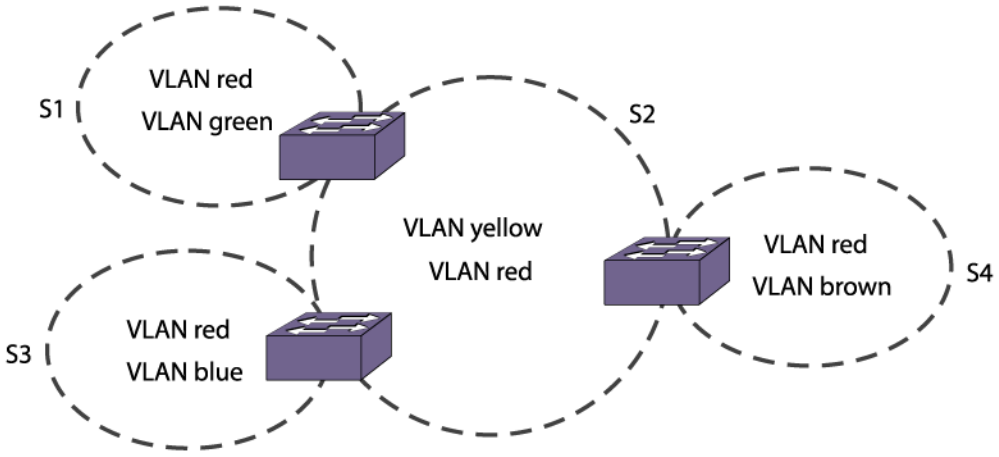


Figure 230: EMISTP Configuration Example

Note
By default, all ports added to a user-defined *STPD* or STPD s0 are in dot1d mode, unless otherwise specified.

The following commands configure the switch located between S1 and S2:

```
create vlan red
configure red tag 100
configure red add ports 1:1-1:4 tagged
create vlan green
configure green tag 200
configure green add ports 1:1-1:2 tagged
```

```
create vlan yellow
configure yellow tag 300
configure yellow add ports 1:3-1:4 tagged
create stpd s1
configure stpd s1 add green ports all emistp
configure stpd s1 add red ports 1:1-1:2 emistp
configure stpd s1 tag 100
enable stpd s1
create stpd s2
configure stpd s2 add yellow ports all emistp
configure stpd s2 tag 300
configure stpd s2 add red ports 1:3-1:4 emistp
enable stpd s2
```

RSTP 802.1w Configuration Example

Figure 231 on page 1850 is an example of a network with multiple STPDs that can benefit from RSTP.

For RSTP to work:

1. Create an STPD.
2. Configure the mode of operation for the STPD.
3. Create the VLANs and assign the VLAN ID and the VLAN ports.
4. Assign the carrier VLAN.
5. Add the protected VLANs to the STPD.
6. Configure the port link types.
7. Enable STP.

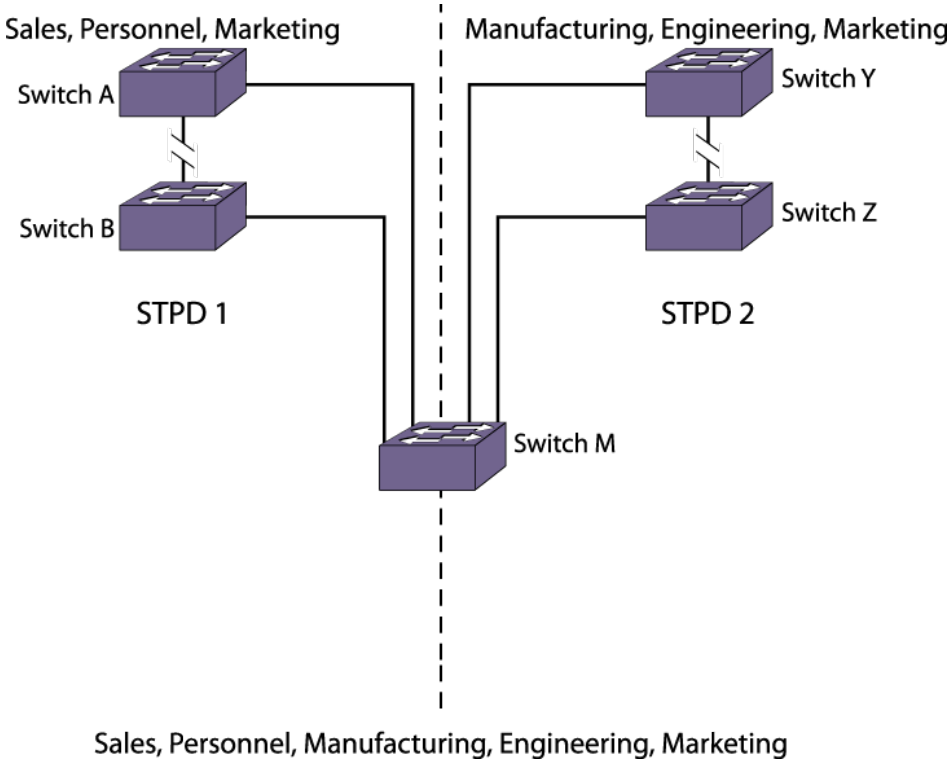


Figure 231: RSTP Example

In this example, the commands configure Switch A in STPD1 for rapid reconvergence.

Use the same commands to configure each switch and STPD in the network.

```
create stpd stpd1
configure stpd stpd1 mode dot1w
create vlan sales
create vlan personnel
create vlan marketing
configure vlan sales tag 100
configure vlan personnel tag 200
configure vlan marketing tag 300
configure vlan sales add ports 1:1,2:1 tagged
configure vlan personnel add ports 1:1,2:1 tagged
configure vlan marketing add ports 1:1,2:1 tagged
configure stpd stpd1 add vlan sales ports all
configure stpd stpd1 add vlan personnel ports all
configure stpd stpd1 add vlan marketing ports all
configure stpd stpd1 ports link-type point-to-point 1:1,2:1
configure stpd stpd1 tag 100
enable stpd stpd1
```

MSTP Configuration Example

You must first configure a CIST before configuring any MSTIs in the region. You cannot delete or disable a CIST if any of the MSTIs are active in the system.

Figure 232 is an example with multiple *STPDs* that can benefit from *MSTP*. In this example, we have two MSTP regions that connect to each other and one external 802.1w bridge.

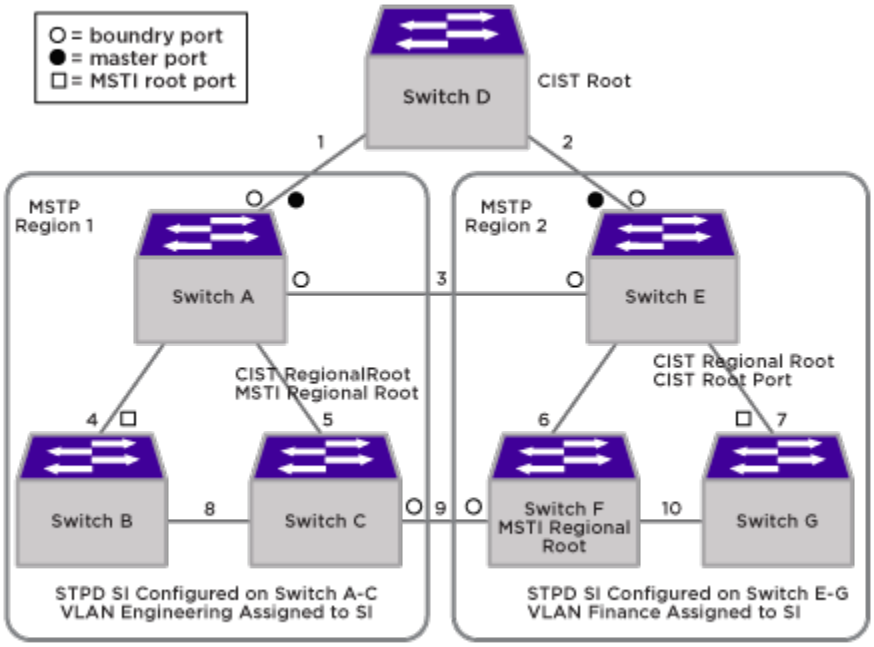


Figure 232: MSTP Configuration Example

For MSTP to work, complete the following steps on all switches in Region 1 and Region 2:

1. Remove ports from the VLAN Default that will be added to VLAN Engineering.
2. Create the VLAN Engineering.
3. Assign a VLAN ID to the VLAN Engineering.

   Note
   If you do not explicitly configure the VLAN ID in your MSTP deployment, use the `show vlan` command to see the internal VLAN ID automatically assigned by the switch.

4. Add ports to the VLAN Engineering.
5. Create the MSTP region.

   Note
   You can configure only one MSTP region on the switch at any given time.

6. Create the STPD to be used as the CIST, and configure the mode of operation for the STPD.
7. Specify the priority for the CIST.
8. Enable the CIST.
9. Create the STPD to be used as an *MSTI* and configure the mode of operation for the STPD.
10. Specify the priority for the MSTI.
11. Assign the VLAN Engineering to the MSTI.
12. Configure the port link type.
13. Enable the MSTI.

On the external switch (the switch that is not in a region):

1. Create an STPD that has the same name as the CIST, and configure the mode of operation for the STPD.
2. Specify the priority of the STPD.
3. Enable the STPD.

Note
In the following sample configurations, any lines marked (Default) represent default settings and do not need to be explicitly configured. STPD s0 already exists on the switch.

In the following example, the commands configure Switch A in Region 1 for MSTP. Use the same commands to configure each switch in Region 1:

```
create vlan engineering
configure vlan engineering tag 2
configure vlan engineering add port 2-3 tagged
configure mstp region region1
create stpd s0 (Default)
disable stpd s0 auto-bind vlan Default
configure stpd s0 mode mstp cist (Default)
configure stpd s0 priority 32768 (Default)
enable stpd s0 (Default)
```

```
create stpd s1
configure stpd s1 mode mstp msti 1
configure stpd s1 priority 32768 (Default)
enable stpd s1 auto-bind vlan engineering
configure stpd s0 ports link-type point-to-point 2-3
enable stpd s1
```

In the following example, the commands configure Switch E in Region 2 for MSTP. Use the same commands to configure each switch in Region 2:

```
create vlan finance
configure vlan finance tag 2
configure vlan finance add port 2-3 tagged
configure mstp region region2
create stpd s0 (Default)
configure stpd s0 mode mstp cist (Default)
configure stpd s0 priority 32768 (Default)
disable stpd s0 auto-bind vlan Default
enable stpd s0 (Default)
create stpd s1
configure stpd s1 mode mstp msti 1
configure stpd s1 priority 32768 (Default)
enable stpd s1 auto-bind vlan finance
configure stpd s0 ports link-type point-to-point 2-3
```

In the following example, the commands configure switch D, the external switch. Switch D becomes the CIST root bridge:

```
create stpd s0 (Default)
disable stpd s0
configure stpd s0 mode dot1w
configure stpd s0 priority 28672
enable stpd s0 auto-bind vlan Default (Default)
configure stpd s0 ports link-type point-to-point 4-5
enable stpd s0
```

Supported Standards, Protocols, and MIBs

Extreme Networks Proprietary MIBs on page 1854
MIB Support Details on page 2038

This appendix provides information about the MIB support provided by the ExtremeXOS *SNMP (Simple Network Management Protocol)* agent residing on Extreme Networks devices running ExtremeXOS and Extreme Networks proprietary MIBS.

Note

All information for supported standards and protocols can be found in the following location: http://www.extremenetworks.com/resources. This appendix contains only the MIB support details.

Extreme Networks Proprietary MIBs

The Extreme Networks MIBs are located at http://www.extremenetworks.com/support/policies/mibs/.

EXTREME ACL MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeAclStatsTable	extremeAclStatsVlanIfIndex	The IfIndex of the *VLAN (Virtual LAN)* in which this policy/rule is applied
	extremeAclStatsPortIfIndex	The IfIndex of the the port in which this policy/rule is applied
	extremeAclDirection	The ingress/egress direction to which this policy/rule is applied (0 for ingress and 1 for Egress)
	extremeAclStatsCounterName	Name of the counter for which the stats is requested
	extremeAclStatsPktCount	The total number of packets that matches this rule
	extremeAclStatsByteCount	The total number of bytes that matches this rule

CTRON Alias MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
ctAliasTable		The ctAliasTable contains information about the aliases known by the agent. This table is indexed using a unique and arbitrary ID and a TimeFilter to allow a management station to learn all alias entries that have been modified since time X. This table is NOT designed to do a single GET. This Table is designed for MIB Walks only. This table contains objects that map upper-layer network services and entities to the physical location of the associated port, and the MAC address of the associated end-system (node). All entries in this table are deleted when sysUpTime is reset to zero. This table allows for a time filtered view. It is useful for determining what has been modified in the Alias Table since a specified time. If a management station reads the ctAliasTable of an entity at time X, subsequent updates can be realized by performing get-next requests of this table to determine which entries have been modified since time X. Then the management station can use GET requests to retrieve only the records of interest from the Alias table. There is a delay between retrieving the Alias Reference values from this table, and a GET request to retrieve a modified row from the Alias table. The management station should be prepared for two possibilities: 1) the GET request may fail if the Alias table entry has been deleted during the delay, and 2) the entry may have been modified during the delay, so the modification time of the retrieved entry may differ from the modification time of the Delta entry. If the entry in the Alias Control table identified by ctAliasID is marked inactive by management request, and subsequently deleted to free resources, all corresponding

Table/Group	Supported Variables	Comments
		entries in this table are also deleted. If the agent needs to free resources, it can delete entries from this table. It SHOULD first remove those which have been marked inactive in the ctAliasControlTable, then it SHOULD remove those entries with the oldest modification times. Access = not-accessible
	CtAliasEntry	Each entry contains information pertaining to the alias of a node. Access = not-accessible
	ctAliasTimeFilter	The value of sysUpTime when this entry was last modified. Access = not-accessible
	ctAliasReference	An integer ID assigned by the agent. The value of this object is equal to an existing ctAliasID. Access = not-accessible
	ctAliasInterface	The ifIndex of the interface on which this entry was discovered. The ifIndex is an 'external value' that identifies a physical component associated with the Alias entry. This object can be used to identify the physical location of the interface in the ifTable [RFC2863]. The possibility of ifIndex value re-assignment must be accommodated by a management application whenever the value of sysUpTime is reset to zero. Since ifIndex values in different 'naming scopes' are not related to one another, the interface to physical component associations are relative to the same logical entity within the agent. For more discussion of naming scopes, see the ENTITY-MIB [RFC2037]. Access = read-only
	ctAliasMacAddress	The MAC address associated with this alias. The TEXTUAL-CONVENTION MacAddress is defined in SNMPv2-TC as an 802 MAC address represented in 'canonical' order as defined by IEEE 802.1a, i.e., as if it were transmitted least significant bit first."

Table/Group	Supported Variables	Comments
		Access = read-only
	ctAliasVlanID	The *VLAN* ID contained in the header of the message associated with this alias. On agents which support this MIB but do not support VLANs, the value for this object MUST be (1). Access = read-only
	ctAliasProtocol	The address type of this alias address. Access = read-only
	ctAliasAddress	The address associated with this entry. The exact format of this object will depend upon the protocol type in ctAliasProtocol. The value returned for this object MUST be in a non-textual format. Access = read-only
	ctAliasIsActive	The status of this entry. A true(1) value indicates the alias is active. A false(2) value indicates an entry which has been marked inactive by network management, using the ctAliasEntryStatus object in the ctAliasControlTable. Entries which have been marked inactive may remain in the table to allow consistent mappings between network management stations until an agent needs to free resources. Access = read-only
	ctAliasAddressText	The address associated with this entry in a human readable textual format. Access = read-only

Table/Group	Supported Variables	Comments
ctAliasControlTable		This table allows for control of Alias entries. Currently control is limited to marking entries as marked for deletion. If the entry in the Alias Control table identified by ctAliasID is marked inactive by management request, and subsequently deleted to free resources, all corresponding entries in the ctAlias table are also deleted. All entries in this table are deleted when sysUpTime is reset to zero. Access = not-accessible
	CtAliasControlEntry	Each entry contains a unique identifier of an Alias entry and a control object to mark the entry as 'marked for deletion'. Access = not-accessible

Table/Group	Supported Variables	Comments
	ctAliasID	A unique and arbitrary integer ID assigned by the agent to identify this entry. Each identifier must not be reused until the agent is re-initialized, i.e. sysUpTime is reset to zero. Access = not-accessible
	ctAliasMarkInactive	This object provides the control to mark this entry to inactive. When an entry is created by the agent, this is set to false(2). A network management application can set this value to true(1), indicating that the entry is no longer needed. Once this object has been set to true(1), it cannot be set to false(2). If the alias is detected again by the agent, the agent may create a new entry for the Alias. When an agent needs to free up resources, it may choose to delete those entries marked inactive. Access = read/write
	ctAliasEntryStatus	This object provides the control to remove or inactivate entries that are no longer needed. active(1) - indicates that this entry is active. inactive(2) - indicates that this entry is no longer of any interest and can be removed, as needed, by the entity. If the alias is marked inactive and is seen again by the agent, the alias ID's status will be changed to active. remove(3) - indicates that the agent SHOULD immediately remove the entry from the table. If the alias is seen again by the agent after having been removed, it will be added as a new aliasID. Access = read/write

Table/Group	Supported Variables	Comments
	ctAliasTableStatsTotalEntries	The total number of entries in the ctAliasControlTable. Access = read-only
	ctAliasTableStatsActiveEntries	The total number of active entries in the ctAliasControlTable. Access = read-only
	ctAliasTableStatsPurgeTime	System time of the most recent purge. Access = read-only
	ctAliasTableStatsState	Current Alias Table state. notStarted(1) indicates the agent has not completed any required start up routines and therefore the ctAliasTable and ctAliasControlTable are not currently valid. ready(2) indicates that the agent has completed all initialization procedures and that the alias MIB is running properly. full(3) indicates that due to memory restrictions on the agent, the ctaliasTable has reached the maximum number of entries possible. NOTE: If an agent is using a circular queue design, then the state of full(3) will never be set. Access = read-only
	ctAliasConfigurationSystemAllocatedEntries	The total number of alias entries allocated. Access = read-only
	ctAliasConfigurationSystemTotalEntries	The total number of alias entries available to be allocated for this system. Access = read-only

Table/Group	Supported Variables	Comments
ctAliasConfigurationTable		This table is used to prevent any single chatty port from monopolizing all of the available alias buffers in the system by letting an administrator tune each port. Access = not-accessible
	ctAliasConfigurationEntry	Each entry contains an enable/disable control variable and variables showing the number of maximum allowable and currently used entries. Access = not-accessible
	ctAliasConfigurationInterfaceTotalEntries	The total number of alias entries currently in use by this ctAliasInterface. Access = read-only
	ctAliasConfigurationInterfaceMaxEntries	This object represents the maximum number of alias entries in the ctAliasTable which were created by this ctAliasInterface. When the object ctAliasConfigurationInterfaceTotalEntries equals this object, the next alias appearing on this ctAliasInterface causes the oldest entry created by this ctAliasInterface to be deleted. The sum of this object across all ctAliasInterface shall not exceed ctAliasConfigurationSystemTotalEntries. If the value of this object is increased, then oldest entry removal ceases until the maximum is reached again. If management reduces the value of this object, then, starting with the oldest, alias entries are removed until the new number of entries is reached. Access = read/write
	ctAliasConfigurationInterfaceEnableState	The enable state of the alias agent on this ctAliasInterface. Access = read/write
	ctAliasConfigurationNumQueueWraps	This object represents the number of times the queue for this ctAliasConfigurationEntry has wrapped. Since circular queues are used for storing the entries, when all allocated entries are used, the oldest entries are reused, thus creating a wrap condition. A value

Table/Group	Supported Variables	Comments
		of zero indicates the queue has not wrapped, except in the case that the counter itself has wrapped. Access = read-only
	ctAliasConfigurationProtocolEnableState	A bit field of the protocol types where a set indicates the detection of that protocol is enabled. A bit position with a cleared bit indicates detection of that protocol is not enabled. Access = read-only

Table/Group	Supported Variables	Comments
	ctAliasMacAddressInterface	The ifIndex of the interface on which this entry was discovered. The ifIndex is an 'external value' that identifies a physical component associated with the Alias entry. This object can be used to identify the physical location of the interface in the ifTable [RFC2863]. The possibility of ifIndex value re-assignment must be accommodated by a management application whenever the value of sysUpTime is reset to zero. Since ifIndex values in different 'naming scopes' are not related to one another, the interface to physical component associations are relative to the same logical entity within the agent. For more discussion of naming scopes, see the ENTITY-MIB [RFC2037] Access = read-only
	ctAliasMacAddressVlanID	The VLAN ID contained in the header of the message associated with this alias. On agents which support this MIB but do not support VLANs, the value for this object MUST be (1). Access = read-only
	ctAliasMacAddressIsActive	The status of this entry. A true(1) value indicates the alias is active. A false(2) value indicates an entry which has been marked inactive by network management, using the ctAliasEntryStatus object in the ctAliasControlTable. Entries which have been marked inactive may remain in the table to allow consistent mappings between network management stations until an agent needs to free resources. Access = read-only
	ctAliasMacAddressAddressText	The address associated with this entry in a human readable textual format. Access = read-only

Table/Group	Supported Variables	Comments
	ctAliasMacAddressTime	The value of sysUpTime when this entry was last modified. Access = read-only
	ctAliasProtocolAddressTable	The ctAliasProtocolAddressTable contains information about the aliases known by the agent indexed by protocol and address. This table contains information that is identical to the ctAliasTable, but it is instanced by protocol and address to allow management entities to search for a protocol and address, or part of a protocol and address, using a single SNMP GetNext request. Access = not-accessible
	ctAliasProtocolAddressEntry	Each entry contains information pertaining to the alias of a node in a table indexed by protocol type, protocol address, MAC address, and alias ID. Access = not-accessible

Table/Group	Supported Variables	Comments
	ctAliasProtocolAddressInterface	The ifIndex of the interface on which this entry was discovered. The ifIndex is an 'external value' that identifies a physical component associated with the Alias entry. This object can be used to identify the physical location of the interface in the ifTable [RFC2863]. The possibility of ifIndex value re-assignment must be accommodated by a management application whenever the value of sysUpTime is reset to zero. Since ifIndex values in different 'naming scopes' are not related to one another, the interface to physical component associations are relative to the same logical entity within the agent. For more discussion of naming scopes, see the ENTITY-MIB [RFC2037] Access = read-only
	ctAliasProtocolAddressVlanID	The VLAN ID contained in the header of the message associated with this alias. On agents which support this MIB but do not support VLANs, the value for this object MUST be (1). Access = read-only
	ctAliasProtocolAddressIsActive	The status of this entry. A true(1) value indicates the alias is active. A false(2) value indicates an entry which has been marked inactive by network management, using the ctAliasEntryStatus object in the ctAliasControlTable. Entries which have been marked inactive may remain in the table to allow consistent mappings between network management stations until an agent needs to free resources. Access = read-only
	ctAliasProtocolAddressAddressText	The address associated with this entry in a human readable textual format. Access = read-only

Table/Group	Supported Variables	Comments
	ctAliasProtocolAddressTime	The value of sysUpTime when this entry was last modified. Access = read-only
	ctAliasEntryClearAll	This object is used to remove all of the alias entries from all of the tables. When set to true(1) all known aliases will be immediately removed from all of their respective tables. A read will always return false(2). Access = read/write
ctAliasInterfaceTable		The ctAliasInterfaceTable contains information about the aliases known by the agent indexed by interface address and alias ID. This table contains information that is identical to the ctAliasTable, but it is instanced by interface address and alias ID to allow management entities to search for an interface address, using a single SNMP GetNext request. Access = not-accessible
	ctAliasInterfaceEntry	Each entry contains information pertaining to the alias of a node in a table indexed by interface address and alias ID. Access = not-accessible
	ctAliasInterfaceMacAddress	The MAC address associated with this alias. The TEXTUAL-CONVENTION MacAddress is defined in SNMPv2-TC as an 802 MAC address represented in 'canonical' order as defined by IEEE 802.1a, i.e., as if it were transmitted least significant bit first. Access = read-only
	ctAliasInterfaceAddress	The address associated with this entry. The exact format of this object will depend upon the protocol type in ctAliasInterfaceProtocol. The value returned for this object MUST be in a non-textual format. Access = read-only

Table/Group	Supported Variables	Comments
	ctAliasInterfaceVlanID	The VLAN ID contained in the header of the message associated with this alias. On agents which support this MIB but do not support VLANs, the value for this object MUST be (1). Access = read-only
	ctAliasInterfaceIsActive	The status of this entry. A true(1) value indicates the alias is active. A false(2) value indicates an entry which has been marked inactive by network management, using the ctAliasEntryStatus object in the ctAliasControlTable. Entries which have been marked inactive may remain in the table to allow consistent mappings between network management stations until an agent needs to free resources. Access = read-only
	ctAliasInterfaceAddressText	The address associated with this entry in a human readable textual format. Access = read-only
	ctAliasInterfaceTime	The value of sysUpTime when this entry was last modified. Access = read-only
ctAliasMacAddressTable		The ctAliasMacAddressTable contains information about the aliases known by the agent indexed by MAC address and protocol. This table contains information that is identical to the ctAliasTable, but it is instanced by MAC address and protocol to allow management entities to search for a MAC address and protocol, or part of a MAC address and protocol, using a single *SNMP* GetNext request. Access = not-accessible
	ctAliasMacAddressEntry	Each entry contains information pertaining to the alias of a node in a table indexed by MAC address, protocol type, protocol address, and alias ID. Access = not-accessible

Table/Group	Supported Variables	Comments
ctAliasBasicGroup		A collection of objects for the discovery and mapping of end systems and aliases.
ctAliasStatsGroup		A collection of objects pertaining to discovery and mapping of end systems and aliases.
ctAliasControlGroup		This group has been deprecated. See ctAliasControlGroupI.
ctAliasConfigurationGroup		A collection of objects for configuring port buffer queues.
ctAliasMacAddressGroup		A collection of objects for the discovery and mapping of end systems and aliases.
ctAliasProtocolAddressGroup		A collection of objects for the discovery and mapping of end systems and aliases.
ctAliasControlGroupI		A collection of objects which make the Aliases accessible using the TimeFilter TEXTUAL CONVENTION.
ctAliasGroup		An object used for removing alias entries.
ctAliasConfigurationGroupI		An object to indicate queue usage.
ctAliasConfigurationGroup2		A collection of objects for configuring port buffer queues.
ctAliasInterfaceGroup		A collection of objects for the discovery and mapping of end systems and aliases.

EXTREME-CFGMGMT-MIB

This MIB provides the ability to query configuration activity and to perform switch management operations. The following features are part of this MIB:

- Expose the time and date of the last configuration save operation.
- Issue a notification to NMS upon completion of a database save operation.
- Provide the ability for ExtremeXOS to report on-demand the presence of unsaved configuration changes on the system.
- Issue a notification whenever any system configuration is altered, added, or deleted.

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeLastSaveCfgTable	extremeLastSavedEntry	This group of objects contain of the last saved configuration operation for the running con If the value of extremeLastChangeConfigTin is greater than extremeLastSaveConfigTime, configuration has been chang not saved.
	extremeLastSaveConfigSlotId	Specifies the MSM/MM modul number of a switch or the slot number currently being used active stack node.
	extremeLastSaveConfigTime	This object return the date an time of the last configuration operation. This data reflects th command or the last setting o extremeSaveConfiguration ob
	extremeLastSaveConfigFileName	This object returns the configu file name of the last save oper
	extremeLastSaveConfigSource	This object indicates the sourc of the last configuration save operation for the configuratio file associated with this entry, extremeLastSaveConfigFileNa

Table/Group	Supported Variables	Comments
extremeLastChangeCfgTable	extremeLastChangeEntry	This group of objects contain of the last change configurati operation for the running con If the value of extremeLastChangeConfigTin is greater than extremeLastSaveConfigTime, configuration has been chang not saved
	extremeLastChangeConfigSlotId	Specifies the slot number curr being used by the active stack
	extremeLastChangeConfigTime	This object returns the date ar time of the last configuration change operation. This data re the last command or the last s of the extremeSaveConfigurat object.
	extremeLastChangeConfigFileName	This object returns the configu file name of the last change operation.
	extremeLastChangeConfigSource	This object indicates the source of the last configuration change operatio extremeLastChangeConfigFile
cfgMgmtTraps	cfgMgmtConfigSaveTrap	This notification indicates that the running configuration of t managed system has updated NVRAM storage of the current configuration.
	cfgMgmtConfigChangeTrap	This notification contain detai of the last change configurati operation for the running con
cfgMgmtControl	cfgMgmtConfigSaveTrapEnable	This variable indicates whether the system produces the cfgMgmtConfigSaveTrap notification. A false value prev notifications from being gene by this system.
	cfgMgmtConfigChangeTrapEnable	This variable indicates whether the system produces the cfgMgmtConfigChangeTra notification. A false value prev notifications from being gene by this system.

ENTERASYS-APPLICATION-SIGNATURE-MIB

Table/Group	Supported Variables	Comments
Application Group Table	etsysAppSignGroupTable	A table that contains the standard Application Groups for the Extreme Network deep packet inspection engine. Not-accessible.
	etsysAppSignGroupEntry	An entry that ties a given application Group Name to a particular application group ID. Not-accessible.
	etsysAppSignGroupId	A pre-defined unique group index that points to the group to which the application belongs. Not-accessible.
	etsysAppSignGroupName	A pre-defined application group name. Read-only.

Table/Group	Supported Variables	Comments
Application Display Name Table	etsysAppSignDisplayNameTable	A table that describes attributes of a user-defined application display name. Not-accessible.
	etsysAppSignDisplayNameEntry	Describes a particular entry within the etsysAppSignDisplayNameTable." Not-accessible.
	etsysAppSignDisplayId	A unique ID assigned to a specific extended signature. The user can only assign an index in a range from 5,000 to 5,999. The rest of the indices are reserved for internal use. Not-accessible.
	etsysAppSignDisplayName	A user-friendly name assigned to this extended signature. The display name and display ID *must* be unique within the scope of a single managed entity. This object *must not* be modifiable while this entry's RowStatus is active (1) or if etsysAppSignDisplayId is a reserved index. Read-create.
	etsysAppSignDisplayNameRowStatus	This object allows for the dynamic creation and deletion of entries within the etsysAppSignDisplayNameTable as well as the activation and deactivation of these entries. Read-create.

Table/Group	Supported Variables	Comments
Application Pattern String Table	etsysAppSignPatternTable	A table that describes attributes of a user-defined pattern string. Not-accessible.
	etsysAppSignPatternEntry	An entry describing the matching pattern of a WebApplRule and its assigned index. For use by the Extreme Networks deep packet inspection engine. Not-accessible.
	etsysAppSignPatternIndex	A unique identifier for a pattern string. The user can only assign an index in a range from 1,000,000 to 2,000,000. The rest of the indices are reserved for internal use. Not-accessible.
	etsysAppSignPattern	A case-sensitive string for matching against the 'host' field of an HTTP request packet, or the 'server name' in the 'client Hello' packet of a TLS session. This is used to identify the application. Read-create.
	etsysAppSignPatternRowStatus	This object allows the dynamic creation and deletion of entries within the etsysAppSignPatternTable, as well as the activation and deactivation of these entries. etsysAppSignPatternRowStatus *must not* be set to active (1) and createAndGo(4) if the etsysAppSignPattern object contains an empty string. Read-create.

ENTERASYS-CONVERGENCE-END-POINT-MIB

Table/Group	Supported Variables	Comments
etsysConvEndPointGlobalConfigEnable		When set to enabled(1) this object activates the current Convergence End Point configuration. When set to disabled(2) this MIB SHALL be configurable, but MUST NOT be applied, all entries will be removed from the etsysConvEndPointConnMacTable and the etsysConvEndPointConnPortTable, and any policies that have been changed as a result of End Point detection will revert to their default states. Maintaining the value of this object across agent reboots is REQUIRED. Read-write.
etsysConvEndPointGlobalConfigTable		This table represents all Convergence End Point types supported by this entity, and provides a default policy profile for each Convergence End Point type. Not-accessible.
	etsysConvEndPointGlobalConfigEntry	Defines the default policy to be applied when this type of Convergence End Point is detected or discovered. Not-accessible.
	etsysConvEndPointGlobalDefaultPolicyIndex	The policy that should be applied for this type of Convergence End Point. This value is an etsysPolicyProfileIndex as defined in the ENTERASYS-POLICY-PROFILE-MIB. If no such policy index exists, then no policy will be applied when this type of Convergence End Point is detected or discovered. The value zero is not a valid etsysPolicyProfileIndex, so zero indicates that no policy should be applied. Maintaining the value of this object across system reboots is REQUIRED.

Table/Group	Supported Variables	Comments
		Read-write.
etsysConvEndPointProtocolConfigTable		This table provides control of Convergence End Point detection and discovery on a per interface and per Convergence End Point type basis. Maintaining the values of this table across system reboots is REQUIRED. Not-accessbile.
	etsysConvEndPointProtocolConfigEntry	The current Convergence End Point detection and discovery status for an interface. Not-accessible.
	etsysConvEndPointProtocolEndPointType	The Convergence End Point type. Not-accessible.
	etsysConvEndPointProtocolEnable	When set to enabled(1), Convergence End Point detection and discovery will be enabled on this interface for the Convergence End Point type specified. When set to disabled(2), the Convergence End Points that have been discovered or detected on this interface of the type specified will be removed from the etsysConvEndPointConnMacTable and the etsysConvEndPointConnPortTable. After all of the End Points have been removed from an interface the policy on that interface will revert to its default state. Read-write.
	etsysConvEndPointGlobalConfigAccountEnable	When enabled(1), *RADIUS (Remote Authentication Dial In User Service)* accounting start, interim and stop frames are sent to the configured RADIUS server(s). When disabled(2), accounting packets are not sent to the RADIUS server. Read-write.

Table/Group	Supported Variables	Comments
etsysConvEndPointConnMacTable		This table provides information on each of the Convergence End Points that has been discovered or detected by this entity. Maintaining the value of this object across system reboots is NOT REQUIRED. Not-accessible.
	etsysConvEndPointConnMacEntry	Information about an individual Convergence End Point. Not-accessible.
	etsysConvEndPointConnMacMacAddress	The MAC address of this Convergence End Point. Not-accessible.
	etsysConvEndPointConnMacPortNumber	The interface on which this Convergence End Point was detected or discovered. Read-only.
	etsysConvEndPointConnMacEndPointType	The type of Convergence End Point described by this row. Read-only.
	etsysConvEndPointConnMacPolicyIndex	The policy applied for this Convergence End Point. This value is an etsysPolicyProfileIndex as defined in the ENTERASYS-POLICY-PROFILE-MIB. If the desired policy does not currently exist, or never did exist, then a zero MUST be returned. Read-only.
	etsysConvEndPointConnMacDiscoveryTime	The time and date that this Convergence End Point was detected or discovered. Read-only.
	etsysConvEndPointConnMacFirmwareVersion	This is the firmware version of the discovered Convergence End Point. If this information is not available via the detection protocol then a zero length string MUST be returned. Read-only.
	etsysConvEndPointConnMacInetAddressType	The address type of the discovered or detected Convergence End Point. If the address is unknown,

Table/Group	Supported Variables	Comments
		then this object MUST return 'unknown' Read-only.
	etsysConvEndPointConnMacInetAddress	The address of the discovered or detected Convergence End Point. If the address is unknown, then this object MUST return a zero length string. Read-only.
	etsysConvEndPointConnPortTable	This table provides a mapping from the entities interfaces to discovered Convergence End Points. Maintaining the value of this object across system reboots is NOT REQUIRED. Not-accessible.
	etsysConvEndPointConnPortEntry	Information about an individual Convergence End Point. Not-accessible.
	etsysConvEndPointConnPortMacAddress	The MAC address of the discovered or detected Convergence End Point. Read-only.

Table/Group	Supported Variables	Comments
etsysCEPConnMacTable		This table provides information on each of the Convergence End Points that has been discovered or detected by this entity. Maintaining the value of this object across system reboots is NOT REQUIRED. Not-accessible.
	etsysCEPConnMacEntry	Information about an individual Convergence End Point (CEP). Not-accessible.
	etsysCEPConnMacEndPointType	The type of Convergence End Point described by this row. Read-only.
	etsysCEPConnMacPolicyIndex	The type of Convergence End Point described by this row. Read-only.
	etsysCEPConnMacPolicyIndex	The policy applied for this Convergence End Point. This value is an etsysPolicyProfileIndex as defined in the ENTERASYS-POLICY-PROFILE-MIB. If the desired policy does not currently exist, or never did exist, then a zero MUST be returned. Read-only.
	etsysCEPConnMacDiscoveryTime	The time and date that this Convergence End Point was detected or discovered. Read-only.
	etsysCEPConnMacFirmwareVersion	This is the firmware version of the discovered Convergence End Point. If this information is not available via the detection protocol then a zero length string MUST be returned. Read-only.
	etsysCEPConnMacInetAddressType	The address type of the discovered or detected Convergence End Point. If the address is unknown, then this object MUST return 'unknown'. Read-only.
	etsysCEPConnMacInetAddr	The address of the

Table/Group	Supported Variables	Comments
	ess	discovered or detected Convergence End Point. If the address is unknown, then this object MUST return a zero length string. Read-only.
	etsysConvEndPointDetectionMaxEntries	This value represents the maximum number of entries allowed in the etsysConvEndPointDetectionTable. Read-only.
	etsysConvEndPointDetectionTable	A table that describes the IP address and IP port based Convergence End Point detection parameters. Packets on interfaces that are configured to detect End Points, via the etsysConvEndPointProtocolConfigTable, that are defined in this table are inspected to determine if a Convergence End Point exists on that interface. Rules applying to Convergence End Point types that are configured for detection are evaluated according to the following: 1. If etsysConvEndPointDetectionProtocol does not have any bits set, this indicates that the port based detection should not be performed, proceed to step 2. Otherwise the value in etsysConvEndPointDetectionProtocol is evaluated against the protocol field in the IP header, if it is of a, or the, type specified then the destination port must be between the values of etsysConvEndPointDetectionPortLow and etsysConvEndPointDetectionPortHigh, inclusive. 2. If the etsysConvEndPointDete

Table/Group	Supported Variables	Comments
		ctionAddr is null, this indicates that no detection address check should be performed, consider the End Point detected. Otherwise the value the of etsysConvEndPointDetectionAddr will be masked with the value of etsysConvEndPointDetectionAddrMask, via a bitwise AND operation, and this value will be compared with the destination IP address that has also been masked with the value of etsysConvEndPointDetectionAddrMask, via a bitwise AND operation, if they are equal then consider the End Point detected. If there are no rows in this table that are in the active state then the following pseudo entries describe the behavior of the IP based Convergence End Point detection. EndPointType siemens Protocol { { udp, tcp } } PortLow 4060 PortHigh 4060 AddrType unknown Addr ' ' AddrMaskType unknown AddrMask '' EndPointType h323 Protocol { { udp, tcp } } PortLow 1718 PortHigh 1720 AddrType ipv4 Addr '224.0.1.41' AddrMaskType ipv4 AddrMask '255.255.255.255' EndPointType sip Protocol { { udp, tcp } }

Table/Group	Supported Variables	Comments
		PortLow 5060 PortHigh 5060 AddrType ipv4 Addr '224.0.1.75' AddrMaskType ipv4 AddrMask '255.255.255.255' EndPointType h323 Protocol { { udp, tcp } } PortLow 1718 PortHigh 1720 AddrType unknown Addr '' AddrMaskType unknown AddrMask '' EndPointType sip Protocol { { udp, tcp } } PortLow 5060 PortHigh 5060 AddrType unknown Addr '' AddrMaskType unknown AddrMask '' If one or more entries in this table are active the active entries will define the IP based detection behavior in its entirety. Maintaining entries in this table across system reboots is REQUIRED for entries with an etsysConvEndPointDetectionRowStatus value of active(1). Not-accessible.
	etsysConvEndPointDetectionEntry	Each entry represents a describes parameters for detecting a Convergence End Point. Not-accessible.
	etsysConvEndPointDetectionRuleIndex	The Convergence End Point device type that a 'hit' on this rule will indicate the presence of. Read-create.
	etsysConvEndPointDetectionProtocol	The IP protocol type that the following port range applies to. If no bits are

Table/Group	Supported Variables	Comments
		set then the destination port will not be evaluated as part of the detection process. Read-create.
	etsysConvEndPointDetectionPortLow	The low end of the range of the IP destination port that will be used for Convergence End Point detection. This value represents the low value in a range of port numbers that will be used to indicate this type of Convergence End Point. This value must be less than or equal to etsysConvEndPointDetectionPortHigh. Read-create.
	etsysConvEndPointDetectionPortHigh	The high end of the range of the IP destination port that will be used for Convergence End Point detection. This value represents the high value in a range of port numbers that will be used to indicate this type of Convergence End Point. This value must be greater than or equal to etsysConvEndPointDetectionPortLow. Read-create.
	etsysConvEndPointDetectionAddrType	The destination address type that will be used for Convergence End Point detection. If the address type is 'unknown', then this value will not be evaluated as part of the detection process. Read-create.
	etsysConvEndPointDetectionAddr	The destination IP address that will be used for Convergence End Point detection. If this address is null, then this value will not be evaluated as part of the detection process. Read-create.
	etsysConvEndPointDetectionAddrMaskType	The address type of the mask that will be applied to the destination IP

Table/Group	Supported Variables	Comments
		address as part of the Convergence End Point detection process. If the address type is 'unknown', then this value will not be evaluated as part of the detection process. Read-create.
	etsysConvEndPointDetectionAddrMask	The destination address mask that will be used for Convergence End Point detection. If the mask is null, then this value will not be evaluated as part of the detection process. Read-create.
	etsysConvEndPointDetectionRowStatus	The status of this row as defined by the RowStatus textual convention. Rows in this table can only transition to 'notInService', or to 'active' if the etsysConvEndPointDetectionAddrType object and the etsysConvEndPointDetectionAddrMaskType object have the same value, and either their values are not 'unknown' or there is at least one bit set in the etsysConvEndPointDetectionProtocol object. Read-create.

Table/Group	Supported Variables	Comments
etsysCEPDetectionPortConfigTable		A table of per port configuration for Convergence End Point detection. Maintaining the value of this object across system reboots is NOT REQUIRED. Not-accessible.
	etsysCEPDetectionPortConfigEntry	An entry containing per port Convergence End Point data. Only interfaces that are able to detect Convergence End Points are represented in this table. Not-accessible.
	etsysCEPPortClearUsers	Setting this object to a value of true(1) will cause all detected or discovered Convergence End Points to be removed from the following status tables: etsysConvEndPointConnMacTable etsysConvEndPointConnPortTable etsysCEPConnMacTable Setting this object to a value of false(2) has no effect. This object will always return a value of false(2). Read-write.

Table/Group	Supported Variables	Comments
etsysCEPCdpPortConfigTable		A table of per port configuration for Convergence End Point detection related to the Cisco CDP protocol. Only ports that are capable of participating in Cisco CDP protocol are represented in this table. Not-accessible.
	etsysCEPCdpPortConfigEntry	An entry containing per port Convergence End Point data. Only interfaces that are able to detect Convergence End Points are represented in this table. Not-accessible.
	etsysCEPCiscoCDPVoicePortVlan	This object controls the behavior of the remote configuration of Cisco VOIP phones via Cisco CDP. Setting this object to a value of -1 will cause the device to not attempt to remotely configure a Cisco phone using CDP. Setting this object to a value of 0 will cause the device to configure the phone to send 802.1p priority tagged frames. A value of 4095 will cause the device to configure the phone to send untagged frames. A value in the range 1..4094 will cause the device to configure the phone to send 802.1Q tagged frames using that value as the *VLAN* ID encoded in the tag. This object SHOULD be maintained across device reboots. Read-write.
	etsysConvEndPointGlobalConfigGroup	Objects which provide global control of the Convergence End Point features. Status = deprecated.
	etsysConvEndPointConfigGroup	Objects which provide overrides to the global

Table/Group	Supported Variables	Comments
		Convergence End Point controls. Status = current.
	etsysConvEndPointDetectionGroup	Objects that provide configuration information specific to IP, port, and address based Convergence End Point detection. Status = deprecated.
	etsysConvEndPointStatusGroup	Objects that provide status information for discovered and detected Convergence End Points. Status = deprecated.
	etsysConvEndPointStatusGroup2	Objects that provide status information for discovered and detected Convergence End Points. Status = current.
	etsysConvEndPointDetectionGroup2	Objects that provide configuration information specific to IP, port, and address based Convergence End Point detection Status = current.
	etsysConvEndPointPortConfigGroup	Objects that provide per port control for discovered and detected Convergence End Points. Status = current.
	etsysConvEndPointCiscoVoiceVLANGroup	Objects that provide per port control for configuring Cisco VOIP phones via Cisco CDP. Status = current.
	etsysConvEndPointGlobalConfigGroup2	Objects which provide global control of the Convergence End Point features. Status = current.

EXTREME ENTERASYS ENTITY SENSOR MIB

The Extreme Enterasys Entity Sensor MIB's extremeEntityTempSfpSensorExtEntry contains objects for representing per transceiver sensor information that extends the information provided in the Entity Sensor MIB Extensions' entPhySensorTable of the SFP/QSFP. The following table list the objects supported.

Table/Group	Description	Comments
etsysEntitySfpSensorState	Integer indicates temperature state of sensor	Read-only
etsysEntitySfpSensorHighAlarm	Integer representing the sensor's high alarm threshold	Read-only
etsysEntitySfpSensorHighWarning	Integer representing the sensor's high warning threshold	Read-only
etsysEntitySfpSensorLowAlarm	Integer representing the sensor's low alarm threshold	Read-only
etsysEntitySfpSensorLowAlarm	Integer representing the sensor's low warning threshold	Read-only
etsysEntitySfpSensorTrapEnable	Indicates if notification will be sent when sensor state changes as defined by etsysEntitySfpSensorState	Read-Write

ENTERASYS-MAC-AUTHENTICATION-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
etsysMACAuthenticationSystemGroup	etsysMACAuthenticationSystemEnable	When enabled(1), all objects in this MIB are fully active. When disabled(2), this object overrides all other object settings in this MIB without affecting their values.
	etsysMACAuthenticationMACUserPassword	This is the string to be used as a password credential when authenticating a MAC address when etsysMACAuthenticationMode is set to sharedSecret(1).
	etsysMACAuthenticationPortUserNameSignificantBits	This object represents the number of significant bits in the MAC addresses to be used starting with the left-most bit of the vendor portion of the MAC address. The significant portion of the MAC address is sent as a user-name credential when the primary attempt to authenticate the full MAC address fails. Any other failure to authenticate the full address, (i.e. authentication server timeout) causes the the next attempt to start once again with a full MAC authentication."
	etsysMACAuthenticationMode	This object is used to determine the type of password credential to use when authenticating a MAC address. password(1) - Attempt to authenticate a user with the password credential provided by etsysMACAuthenticationMACUserPassword. radiusUsername(2) - Attempt to authenticate a user with a password credential that is the same as their radius username credential.
	etsysMACAuthenticationSystemAccountEnable	When enabled(1), RADIUS accounting start, interim and stop frames are sent to the configured RADIUS server(s).

Table/Group	Supported Variables	Comments
		When disabled(2), accounting packets are not sent to the RADIUS server.
	etsysMACAuthentication SystemUserNameCase	"When LOWER(1), the user-name credential is the MAC address formatted as xx:xx:xx:xx:xx:xx or xx-xx-xx-xx-xx-xx or xxxxxxxxxxxx. When set to UPPER(2), the user-name credential is the MAC address formatted as XX:XX:XX:XX:XX:XX or XX-XX-XX-XX-XX-XX or XXXXXXXXXXXX." DEFVAL { upper } ::= { etsysMACAuthenticationSystem 7 } etsysMACAuthenticationSystemGroup4 OBJECT-GROUP MAX-ACCESS read-write.

Table/Group	Supported Variables	Comments
etsysMACAuthenticationPortConfigGroup	etsysMACAuthenticationPortConfigTable	A table containing configuration objects for each MAC authentication port. The configuration for each port in this table must be non-volatile.
	etsysMACAuthenticationPortConfigEntry	Each conceptual row provides control over all of the initial values used by each authenticated MAC on this port. Subsequent changes to rows in this table, except where noted, have no effect on existing MACs authenticated on this port.
	etsysMACAuthenticationPort	This is the InterfaceIndex associated with this row.
	etsysMACAuthenticationPortInitialize	When set to true(1), the MAC authentication logic on this port is initialized, forcibly ending all MAC authentication sessions currently in existence on this port. A set with the value false(2) has no affect and a read always returns false.
	etsysMACAuthenticationPortReauthenticate	When set to true(1), the MAC authentication entity on this port is required to immediately verify all currently authenticated MACs on this port. This requires that each MAC address be authenticated with the authentication server through the local authentication client or some other authentication mechanism. Each supplicant remains authenticated pending the outcome.
	etsysMACAuthenticationPortEnable	When set to enabled(1), a platform dependent triggering mechanism initiates an authentication exchange using a MAC address for authentication credentials. When disabled(2), authentication attempts are disabled and all currently authenticated MAC sessions or those in the process of authentication on this port are terminated.
	etsysMACAuthenticationPortQuietPeriod	Unsupported object.
	etsysMACAuthenticationPortReauthPeriod	The value, in seconds, between attempts to re-authenticate any current MAC authenticated on this port.
	etsysMACAuthentication	If enabled(1), then every

Table/Group	Supported Variables	Comments
	PortReauthEnabled	etsysMACAuthenticationReauthPeriod the switch attempts to validate all currently authenticated MACs on this port. When set to disabled(2) all current re-authentications in progress are allowed to complete and the requisite actions are taken. When set to disabled(2), no further re-authentications are attempted.
	etsysMACAuthentication AuthenticationsAllowed	The maximum number of concurrent authentications supported on this port on this module. The default value of this object is platform and resource dependent.
	etsysMACAuthentication AuthenticationsAllocated	The maximum number of MAC authentications permitted on this port on this module. This value must be non-zero and be less than or equal to the value of etsysMACAuthenticationAuthenticationsAllowed. Setting this object to a value less than the current number of authenticated MACs on this port prevents further authentications, but has no affect on the current sessions.
	etsysMACAuthentication LastFailedAuthCause	The string will be formatted with 'XX-XX-XX-XX-XX-XX: TIME&DATE: Textual failure reason'; where XX-XX-XX-XX-XX-XX is the MAC address and TIME&DATE is the time (hh/mm/ss) and date (mm/dd/yyyy) of the failure. It is also only best effort; as there could be multiple failures per port and the agent may query this at any random time.

Table/Group	Supported Variables	Comments
etsysMACAuthenticationMACConfigGroup	etsysMACAuthenticationMACConfigTable	A table containing configuration objects for each MAC authenticated on a port. Each row in this table is created dynamically when a MAC authenticates on a port.
	etsysMACAuthenticationMACConfigEntry	Each conceptual row inherits it's initial information from the row in the etsysMACAuthenticationPortConfigTable corresponding to the correct port. Each row represents an authenticated MAC.
	etsysMACAuthenticationMACAddress	This is the MAC address that was authenticated on this port.
	etsysMACAuthenticationSupplicantPort	This is the InterfaceIndex associated with this rows authenticated MAC.
	etsysMACAuthenticationMACInitialize	When set to true(1), this MAC session terminates causing the corresponding row in this table and in the etsysMACAuthenticationSessionTable to be removed. Setting this object to false(2) has no effect on the system. Reads of this object always return false(2).
	etsysMACAuthenticationMACReauthenticate	When set to true(1), this MAC authentication session on this port is required to immediately verify it's credentials. This requires that each MAC address be authenticated with the authentication server through the local authentication client or some other authentication mechanism. Setting this object to false(2) has no effect on the system. Reads of this object always return false(2).
	etsysMACAuthenticationMACReauthPeriod	The value, in seconds, between attempts to re-authenticate the MAC associated with this row.
	etsysMACAuthenticationMACReauthEnabled	If enabled(1), then every etsysMACAuthenticationReauthPeriod the switch attempts to validate all currently authenticated MACs on this port. If disabled(2), reauthentication is not attempted.

Table/Group	Supported Variables	Comments
etsysMACAuthenticationSessionGroup	etsysMACAuthenticationSessionTable	A table containing configuration objects for each MAC authentication on a port. The successful completion of an authentication causes the creation of a new row in this table. When a MAC becomes unauthenticated because of a link-down, a management change, or system re-initialization, then the corresponding row is removed from this table.
	etsysMACAuthenticationSessionEntry	Each conceptual row inherits it's initial information from the row in the etsysMACAuthenticationPortConfigTable corresponding to the correct port. Each row represents an authenticated MAC.
	etsysMACAuthenticationSessionPort	This is the InterfaceIndex associated with the authenticated MACs session.
	etsysMACAuthenticationDuration	The value, in seconds, which have elapsed since the start of this session.

ENTERASYS-MIRROR-CONFIG-MIB

The following tables, groups, and variables are supported in this MIB.

Table 163: ENTERASYS-MIRROR-CONFIG-MIB

Table/Group	Supported Variables	Comments
etsysMirrorSystemMaxLocalMirrors		This value represents the maximum number of local mirrors that the device supports. Read-only
etsysMirrorSystemMaxRemoteMirrors		This value represents the maximum number of remote mirrors that the device supports. Read-only
etsysMirrorSystemMaxLocalMirrorDestinationPorts		This value represents the maximum number of destination ports supported for a local mirror. Read-only

Table 163: ENTERASYS-MIRROR-CONFIG-MIB (continued)

Table/Group	Supported Variables	Comments
etsysMirrorSystemMaxMirrorDestinationControlGroups		This value represents the maximum number of rows in the etsysMirrorDestinationControlTable. Read-only
etsysMirrorSystemMaxMirrorFirstN		This value represents the maximum configurable number of initial packets of a flow that can be mirrored. If this value is 0, this feature is not supported. Read-only

Table 163: ENTERASYS-MIRROR-CONFIG-MIB (continued)

Table/Group	Supported Variables	Comments
Mirror Table	etsysMirrorDestinationNextAvailableIndex	This object indicates the numerically lowest available index within this entity, which may be used for the value of etsysMirrorDestinationControlIndex in the creation of a new entry in the etsysMirrorDestinationControlTable. An index is considered available if the index value falls within the range of 1 to (newLeaf with max dest groups) and is not being used to index an existing entry in the etsysMirrorDestinationControlTable contained within this entity. A value of zero indicates that all of the entries in the etsysMirrorDestinationControlTable are currently in use. This value SHOULD only be considered a guideline for management creation of etsysMirrorDestinationControlTable entries, there is no requirement on management to create entries based upon this index value. Read-only
	etsysMirrorDestinationControlTable	Controls the setup of mirror destination ports on the managed device. Not-accessible
	etsysMirrorDestinationControlEntry	A conceptual row in the etsysMirrorDestinationControlTable. Not-accessible
	etsysMirrorDestinationControlIndex	A unique arbitrary index for this etsysMirrorDestinationControlEntry. This object MUST have a value between 1 and etsysMirrorSystemMaxMirrorDestinationControlGroups. Not-accessible
	etsysMirrorDestinationCont	Administratively assigned

Table 163: ENTERASYS-MIRROR-CONFIG-MIB (continued)

Table/Group	Supported Variables	Comments
	rolOwner	name of the owner of this entry. It usually defines the entity that created this entry and is therefore using the resources assigned to it, though there is no enforcement mechanism, nor assurance that rows created are ever used. Read-create
	etsysMirrorDestinationControlStorageType	The memory realization of the conceptual row. volatile(2) -- is lost upon reboot. nonVolatile(3) -- is persistent across reboots. permanent(4) -- there are no writable objects. readOnly(5) -- agent created rows. Read-create
	etsysMirrorDestinationControlRowStatus	This object manages the creation and deletion of this entry. active(1) -- Indicates that this entry is available for use by the managed device. notInService(2) -- Indicates that this entry exists in the agent but is not available for use by the managed device. notReady(3) -- Entry is missing information in order to be available for use by the managed device. createAndGo(4) -- A new entry will be created in this table and the new entry will transition to the active state. createAndWait(5) -- A new entry will be created in this table and the new entry will transition to the notInService state. destroy(6) -- Deletes the row and any associated EtsysMirrorDestinationEntry rows. Read-create
	etsysMirrorDestinationCont	When this object is set to a

Table 163: ENTERASYS-MIRROR-CONFIG-MIB (continued)

Table/Group	Supported Variables	Comments
	rolMirrorFirstN	nonzero value, only the first N packets for a particular flow assigned to this index will be mirrored. Otherwise all packets from the flow are to be mirrored. The maximum value for this object is specified by the etsysMirrorSystemMaxMirrorFirstN object. Read-create
	etsysMirrorDestinationPortTable	Controls the configuration of mirror destination ports. Not-accessible
	etsysMirrorDestinationPortEntry	A conceptual row in the etsysMirrorDestinationPort Table . Not-accessible
	etsysMirrorDestinationPortStorageType	The memory realization of the conceptual row. volatile(2) -- is lost upon reboot. nonVolatile(3) -- is persistent across reboots. permanent(4) -- there are no writable objects. readOnly(5) -- agent created rows. Read-create
	etsysMirrorDestinationPortRowStatus	This object manages the creation and deletion of this entry. active(1) -- Indicates that this entry is available for use by the managed device. notInService(2) -- Indicates that this entry exists in the agent but is not available for use by the managed device. notReady(3) -- Entry is missing information in order to be available for use by the managed device. A transition to this state will happen when the etsysMirrorDestinationIfIndex object for this entry is 0 (zero).

Table 163: ENTERASYS-MIRROR-CONFIG-MIB (continued)

Table/Group	Supported Variables	Comments
		createAndGo(4) -- A new entry will be created in this table and the new entry will transition to the active state. createAndWait(5) -- A new entry will be created in this table and the new entry will transition to the notInService state. destroy(6) -- Deletes this row. Read-create

ENTERASYS-MULTI-AUTH-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
Multiple Authentication System Group	etsysMultiAuthSystemSupportedTypes	This object specifies that authentication types that the device supports. A bit will be set for each corresponding type that is supported.
	etsysMultiAuthSystemMaxNumUsers	The maximum number of users the can be actively authenticated or have authentications in progress at one time in the system.
	etsysMultiAuthSystemCurrentNumUsers	The current number of users the are actively authenticated, have authentications in progress, or the device is keeping authentication termination information for in the system.
	etsysMultiAuthSystemMode	The value strictIeee8021x(1) will cause the device to authenticate in strict adherence to IEEE Std. 802.1X-2001. In this mode no other authentication mechanisms will be active. While in this mode, changes may be made to other objects in the MIB, but they will have no effect on the operation of the device until such time as the system mode is changed to etsysMultiAuth(2). A set of this object to a value of etsysMultiAuth(2) will cause the device to authenticate using multiple authenticators simultaneously.
	etsysMultiAuthSystemDefaultPrecedence	The precedence that authentication results will be applied to network traffic by default. This object will have a size equal to the number of enumerations specified by the EtsysMultiAuthTypes textual convention.
	etsysMultiAuthSystemAdminPrecedence	This object allows one to modify the default precedence by which authentication results will be applied to network traffic. Sets to this object are not required to specify all of the types that the device supports. If less types are specified than are supported, then all types that were not specified

Table/Group	Supported Variables	Comments
		will be given an operational precedence based on that type's default precedence relative to the last type specified. For example, if the default precedence is '030102'H and the object is set to '02'H then operational precedence would be '020301'H. A set to this object of a zero length octet string will clear the administrative precedence. In this case the operational precedence would be equal to the default precedence.
	etsysMultiAuthSystemOperPrecedence	This object returns the operational precedence of authentication types as they will be applied to network traffic. The value returned by this object is the calculated result of the etsysMultiAuthSystemDefaultPrecedence and etsysMultiAuthSystemAdminPrecedence objects. This object will have a size equal to the number of enumerations specified by the EtsysMultiAuthTypes textual convention.
	etsysMultiAuthTypePropertiesTable	A table of properties per authentication type.
	etsysMultiAuthTypePropertiesEntry	An entry containing per authentication type properties.
	etsysMultiAuthType	The authentication type the entry properties pertain to.
	etsysMultiAuthSessionTimeout	The maximum number of seconds an authenticated session may last before termination of the session. A value of zero indicates that no session timeout will be applied. This value MAY be superseded by a session timeout value provided by the authenticating server. For example, if a session is authenticated by a RADIUS server, that server may encode a Session-Timeout Attribute in its authentication response. The operational timeout value of a given authenticated session is specified by the etsysMultiAuthSessionSessionTimeout object.
	etsysMultiAuthIdleTimeo	The maximum number of

Table/Group	Supported Variables	Comments
	ut	consecutive seconds an authenticated session may be idle before termination of the session. A value of zero indicates that no idle timeout will be applied. This value MAY be superseded by a idle timeout value provided by the authenticating server. For example, if a session is authenticated by a RADIUS server, that server may encode a Idle-Timeout Attribute in its authentication response. The operational idle timeout value of a given authenticated session is specified by the etsysMultiAuthSessionIdleTimeout object.
	etsysMultiAuthCurrentNumUsers	The current number of users the are actively authenticated or have authentications in progress for this authentication type in the system.
	etsysMultiAuthSystemMaxNumUsersReachedTrapEnable	This object allows for the enabling or disabling the transmission of the etsysMultiAuthSystemMaxNumUsersReached NOTIFICATION.
	etsysMultiAuthSessionsUniquePerPort	When this object is set to true(1) each multi-auth session MAY be unique to the port it was created on. The operational status of this variable can be found using etsysMultiAuthSessionsUniquePerPortOperStatus.
	etsysMultiAuthSessionsUniquePerPortOperStatus	If this object has a value of true(1) each multi-auth session will be unique to the port it was created on. If this object has a value of false(2) each multi-auth session may exist on multiple ports.
	etsysMultiAuthSystemReAuthenticationTimeoutAction	When this object is set to terminate(1) re-authenticating multiauth sessions will be terminated if the re-authentication RADIUS transaction results in a complete timeout. When this object is set to none(2) re-authentication multiauth sessions will be left as they were prior to the re-authentication attempt if the re-

Table/Group	Supported Variables	Comments
		authentication RADIUS transaction results in a complete timeout. A complete timeout occurs when all RADIUS retries to all appropriate RADIUS servers have been exhausted.

Table/Group	Supported Variables	Comments
Multiple Authentication Port Group	etsysMultiAuthPortTable	A table of per port information and configuration for user authentication.
	-etsysMultiAuthPortEntry	An entry containing per port authentication data. Only interfaces that are able to authenticate users are represented in this table.
	etsysMultiAuthPortMode	This object specifies the authorization mode to use for packets received on this interface. A value of forceUnauthorized(1) indicates that the interface is always unauthenticated. A value of forceAuthorized(2) indicates that users on this port will always be considered to be authenticated. A value of authOptional(3) indicates that authentication is optional on this interface. Packets received from unauthenticated users on the interface will be processed using the static configuration of the interface. Users may promote the policy applied to their traffic by actively authenticating on this interface. A value of authRequired(4) indicates that all packets received on the interface will be dropped until authentication succeeds. Some authentication types, such as PWA, will not be fully functional in this mode of operation.
	etsysMultiAuthPortMaxNumUsers	The maximum number of users that can be actively authenticated or have authentications in progress at one time on this interface.
	etsysMultiAuthPortNumUsersAllowed	The user configured number of users that can be actively authenticated or have authentications in progress at one time on this interface. This object has a default value equal to the value of etsysMultiAuthPortMaxNumUsers for this interface. If the value set to this object is less than its current value, it will have the same effect as setting

Table/Group	Supported Variables	Comments
		the etsysMultiAuthPortClearUsers object to a value of true(1).
	etsysMultiAuthPortCurrentNumUsers	The current number of users that are actively authenticated or have authentications in progress at one time on this interface. By definition this value can not exceed the value specified by etsysMultiAuthPortMaxNumUsers for the same interface.
	etsysMultiAuthPortClearUsers	Setting this object to a value of true(1) will cause all users that are currently authenticated or that have authentications in progress on this interface to become unauthenticated. This will cause any such entries with matching ifIndex values in the etsysMultiAuthSessionStationTable tables to change their authorization status to authTerminated(5) Setting this object to a value of false(2) has no effect. This object will always return a value of false(2).
	etsysMultiAuthPortTrapEnable	This object allows for the enabling or disabling of each trap on a per interface basis. Setting a given bit to a value of 1 allows traps of that type to be sent for events on that interface. Setting a given bit to a value of 0 disallows traps of that type to be sent for events on that interface. The individual bits correlate to specific traps as follows: BIT NOTIFICATION ---------- authSuccessTrap(0) etsysMultiAuthSuccess authFailedTrap(1) etsysMultiAuthFailed authTerminatedTrap(2) etsysMultiAuthTerminated maxNumUsersReachedTrap(3) etsysMultiAuthMaxNumUsersReached
	etsysMultiAuthPortTypeTable	A table of per port, per authentication type information.
	etsysMultiAuthPortTypeEntry	An entry containing per port, per authentication type data. Only interfaces that are able to

Table/Group	Supported Variables	Comments
		authenticate users are represented in this table.
	etsysMultiAuthPortTypeCurrentNumUsers	The current number of users the are actively authenticated or have authentications in progress for this authentication type on the specified port.

Table/Group	Supported Variables	Comments
Multiple Authentication Station GroupMultiple Authentication Session Group	etsysMultiAuthStationTable	A table of station configuration on specific interfaces.
	etsysMultiAuthStationEntry	An entry containing authentication information on a per station, per port basis. Only interfaces that are able to authenticate users are represented in this table.
	etsysMultiAuthStationAddrType	The type of station represented by etsysMultiAuthStationAddr.
	etsysMultiAuthStationAddr	The station address for the authenticated user.
	etsysMultiAuthStationClearUsers	Setting this object to a value of true(1) will cause any users with the specified station address that are currently authenticated or that have authentications in progress to become unauthenticated. This will cause any entries with matching etsysMultiAuthStationAddr values in the etsysMultiAuthSessionStationTable tables to change their authorization status to authTerminated(5). Setting this object to a value of false(2) has no effect. This object will always return a value of false(2).
	etsysMultiAuthSessionStationTable	A table of session information and configuration for user authentication. Entries in this table represent users in various stages of authentication. Entries that do not have a etsysMultiAuthSessionStationAuthStatus value of authSuccess(1) or authInProgress(3) MAY be removed by the agent as required in order to free resources for new user authentications.
	etsysMultiAuthSessionStationEntry	An entry containing authentication information on a per station, per port, per authentication agent type basis. Only interfaces that

Table/Group	Supported Variables	Comments
		are able to authenticate users are represented in this table.
	etsysMultiAuthSessionAgentType	The type of authentication agent for this session.
	etsysMultiAuthSessionStationAuthStatus	The status of authentication for this session.
	etsysMultiAuthSessionAuthAttemptTime	The value of sysUpTime when this session last attempted authorization. For entries that have a value of authInProgress(3) for etsysMultiAuthSessionStationAuthStatus this object MAY return a value of zero.
	etsysMultiAuthSessionAuthServerType	The type of authentication server used to authenticate this session. A value of radius(1) indicates that a RADIUS request and response were attempted in order to authenticate the session. A value of local(2) indicates that the session was authenticated by a local file or configuration on the device itself.
	etsysMultiAuthSessionAuthServerAddrType	The type of data returned by etsysMultiAuthSessionAuthServerAddr. If the etsysMultiAuthSessionAuthServerType leaf for this entry has a value of local(2) then this object MUST return a a value of unknown(0).
	etsysMultiAuthSessionAuthServerAddr	The network address of the authentication server for this session. If the etsysMultiAuthSessionAuthServerType leaf for this entry has a value of local(2) then this object MUST return a zero length string.
	The network address of the authentication server for this session. If the etsysMultiAuthSessionAuthServerType leaf for this entry has a value of local(2) then this object MUST return a zero length string.	The Policy Profile Index returned from the authentication server for this session. The value of zero indicates that no policy will be applied for this session. If the etsysMultiAuthSessionStationAuthStatus object returns a value of authSuccess(1), then a value of zero is the result of the policy not being configured on the authorization server. For all other values of etsysMultiAuthSessionStationAuthStatus a value of zero for this object is the result of authorization

Table/Group	Supported Variables	Comments
		not succeeding or not having completed. All values other than zero are valid Policy Profile Indexes that specify the policy profile the user will receive on this interface. If a given user has been authenticated by multiple authentication types on the same interface the policy that is applied to the user's packets is determined by the precedence of the agents as specified by etsysMultiAuthSystemOperPrecedence. These indexes are suitable for indexing in the ENTERASYS-POLICY-PROFILE-MIB.
	etsysMultiAuthSessionIsApplied	This object indicates whether this entry and the policy index contained within it are actively being applied to traffic matching the interface and station address of this entry. A value of true(1) indicates that this entry is being applied. A value of false(2) indicates that the entry is not being applied. Only one authentication type per interface station address ordered pair may be applied at a single time. The operational precedence of the various authentication types determines which if any type will be applied.
	etsysMultiAuthSessionTerminationTime	The local date and time that the session was terminated. If the session is not in the authTerminated(5) state this object MUST return '00000000'H.
	etsysMultiAuthSessionSessionTimeout	The maximum number of seconds this session may last before automatic termination. A value of zero indicates that no session timeout will be applied. This value MAY be provided by the etsysMultiAuthSessionTimeout object or by the authenticating server.
	etsysMultiAuthSessionIdleTimeout	The maximum number of consecutive seconds this session may be idle before automatic termination. A value of zero indicates that no idle timeout will be applied. This value MAY be provided by the

Table/Group	Supported Variables	Comments
		etsysMultiAuthIdleTimeout object or by the authenticating server.
	etsysMultiAuthSessionDuration	The length of this session in seconds. This object MAY return zero for a session in any state other than authSuccess(1).
	etsysMultiAuthSessionIdleTime	The number of consecutive seconds this session has been idle. This object MAY return zero for a session in any state other than authSuccess(1).
	etsysMultiAuthSessionVlanTunnelAttribute	The VLAN Tunnel Attribute (Tunnel-Group-ID) returned from the authentication server for this session. This value is interpreted as the 12 bit VLAN identifier to be applied to traffic from the session entity. Policy VLAN classification rules have precedence in assigning VLAN, however, in the absence of any applicable rules, this VLAN will be used. If the traffic is already tagged, this VLAN will only be applied if TCI overwrite has been enabled (through Policy or ctDot1qPortReplaceTCI). A value of zero indicates that there is no authenticated VLAN ID for the given session (none was provided by the authentication server). Should a session become unauthenticated this value MUST return zero. A value of 4095 indicates that a the session has been authenticated, but that the VLAN returned could not be applied to the port (possibly because of resource constraints or misconfiguration). The traffic from the session entity will be assigned VLAN through Policy or standard 802.1Q mechanisms.
	etsysMultiAuthSessionClear	Setting this object to a value of true(1) will cause the specified table entry with the specified station address, ifIndex and agent type that are currently authenticated or that have authentications in progress to become unauthenticated and their authorization status changes

Table/Group	Supported Variables	Comments
		to authTerminated(5). Setting this object to a value of false(2) has no effect. This object will always return a value of false(2).
	etsysMultiAuthSessionPortTable	A table of session information and configuration for user authentication. This table represents the information specified in the etsysMultiAuthSessionStationTable with alternate indexing for faster lookups of data on per port basis.
	etsysMultiAuthSessionPortEntry	An entry containing authentication information on a per port, per station, per authentication agent type basis. Only interfaces that are able to authenticate users are represented in this table.
	etsysMultiAuthSessionPortAuthStatus	The status of authentication for this session.
Multiple Authentication Module Group	etsysMultiAuthModuleTable	A table of per module information for user authentication.
	etsysMultiAuthModuleEntry	An entry containing per module authentication data. Only physical indexes with a entPhysicalClass of module(9) are represented in this table. Furthermore, each entity represented in this table must have authentication resources that are separate from every other entity in the table.
	etsysMultiAuthModuleMaxNumUsers	The maximum number of users that can be actively authenticated or have authentications in progress at one time on the specified module.
	etsysMultiAuthModuleCurrentNumUsers	The current number of users that are actively authenticated or have authentications in progress at one time on the specified module. By definition this value can not exceed the value specified by etsysMultiAuthModuleMaxNumUsers for the same module.
	etsysMultiAuthModuleMaxNumUsersReachedTrapEnable	This object allows for the enabling or disabling the transmission of the etsysMultiAuthModuleMaxNumUsersReached NOTIFICATION.

Table/Group	Supported Variables	Comments
Multiple Authentication Counters Group	etsysMultiAuthCounterTable	A table of per station and port user counter data.
	etsysMultiAuthCounterEntry	An entry containing user counter data.
	etsysMultiAuthCounterInboundBytes	The number of bytes of user data received on this interface.
	etsysMultiAuthCounterInboundPackets	The number of packets of user data received on this interface.
	etsysMultiAuthCounterOutboundBytes	The number of bytes of user data transmitted on this interface.
	etsysMultiAuthCounterOutboundPackets	The number of packets of user data transmitted on this interface.
	etsysMultiAuthCounterEnable	This object allows for the enabling or disabling of per-user counters.
Multiple Authentication Notification Group	etsysMultiAuthSuccess NOTIFICATION-TYPE	An etsysMultiAuthSuccess trap signifies that the SNMP entity, acting in an agent role, has successfully authenticated a station on one of its interfaces. The included objects of etsysMultiAuthStationAddrType and etsysMultiAuthStationAddr uniquely identify the station that has been authenticated. The interface that the station was authenticated on is specified by the ifIndex object, and the type of authentication used is to authenticate the station is specified by the etsysMultiAuthSessionAgentType object. This trap will only be generated on interfaces that are in the authOptional(3) or authRequired(4) state.

Table/Group	Supported Variables	Comments
	etsysMultiAuthFailed NOTIFICATION-TYPE	An etsysMultiAuthFailed trap signifies that the SNMP entity, acting in an agent role, has identified a station that attempted and subsequently failed to authenticate on one of its interfaces. The included objects of etsysMultiAuthStationAddrType and etsysMultiAuthStationAddr uniquely identify the station that attempted to authenticate. The interface that the station attempted to authenticate on is specified by the ifIndex object, and the type of authentication attempted is specified by the etsysMultiAuthSessionAgentType object. This trap will only be generated on interfaces that are in the authOptional(3) or authRequired(4) state.
	etsysMultiAuthTerminated NOTIFICATION-TYPE	An etsysMultiAuthTerminated trap signifies that the SNMP entity, acting in an agent role, has terminated the authentication of a station on one of its interfaces. The included objects of etsysMultiAuthStationAddrType and etsysMultiAuthStationAddr uniquely identify the station for which authentication was terminated. The interface that the station was previously authenticated on is specified by the ifIndex object, and the type of authentication that the station was terminated for is specified by the etsysMultiAuthSessionAgentType object. This trap will only be generated on interfaces that are in the authOptional(3) or authRequired(4) state.

Table/Group	Supported Variables	Comments
	etsysMultiAuthMaxNumUsersReached NOTIFICATION-TYPE	An etsysMultiAuthMaxNumUsersReached trap signifies that the SNMP entity, acting in an agent role, has an interface where subsequent to a successful authentication, the number of current sessions on the interface equals the maximum number of sessions allowed for that interface. The interface that the maximum number of sessions has been reached is specified by the ifIndex object.
	etsysMultiAuthModuleMaxNumUsersReached NOTIFICATION-TYPE	An etsysMultiAuthModuleMaxNumUsersReached trap signifies that the SNMP entity, acting in an agent role, has a module where subsequent to a successful authentication, the number of current sessions on the module equals the maximum number of sessions allowed for that module. The module that the maximum number of sessions has been reached is specified by the entPhysicalIndex object.
	etsysMultiAuthSystemMaxNumUsersReached NOTIFICATION-TYPE	An etsysMultiAuthSystemMaxNumUsersReached trap signifies that the SNMP entity, acting in an agent role, where subsequent to a successful authentication, has the number of current sessions on the system equals the maximum number of sessions allowed for that system, .

ENTERASYS-POLICY-PROFILE-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
etsysPolicyProfile group	etsysPolicyProfileMaxEntries	The maximum number of entries allowed in the etsysPolicyProfileTable.
	etsysPolicyProfileNumEntries	The current number of entries in the etsysPolicyProfileTable.
	etsysPolicyProfileLastChange	The sysUpTime at which the etsysPolicyProfileTable was last modified.
	etsysPolicyProfileTableNextAvailableIndex	This object indicates the numerically lowest available index within this entity, which may be used for the value of etsysPolicyProfileIndex in the creation of a new entry in the etsysPolicyProfileTable. An index is considered available if the index value falls within the range of 1 to 65535 and is not being used to index an existing entry in the etsysPolicyProfileTable contained within this entity. This value should only be considered a guideline for management creation of etsysPolicyProfileEntries, there is no requirement on management to create entries based upon this index value.
	etsysPolicyProfileTable	A table containing policy profiles. A policy is a group of classification rules which may be applied on a per user basis, to ports or to stations.
	etsysPolicyProfileEntry	Conceptually defines a particular entry within the etsysPolicyProfileTable. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPolicyProfileIndex	A unique arbitrary identifier for this Policy. Since a policy will be applied to a user regardless of his or her location in the network fabric policy names SHOULD be unique within the entire network fabric. Policy IDs and policy names MUST be

Table/Group	Supported Variables	Comments
		unique within the scope of a single managed entity.
	etsysPolicyProfileName	Administratively assigned textual description of this Policy. This object MUST NOT be modifiable while this entry's RowStatus is active(1).
	etsysPolicyProfileRowStatus	This object allows for the dynamic creation and deletion of entries within the etsysPolicyProfileTable as well as the activation and deactivation of these entries. When this object's value is active(1) the corresponding row's etsysPolicyProfilePortVid, etsysPolicyProfilePriority, and all entries within the etsysPolicyClassificationTable indexed by this row's etsysPolicyProfileIndex are available to be applied to network access ports or stations on the managed entity. All ports corresponding to rows within the etsysPortPolicyProfileTable whose etsysPortPolicyProfileOperID is equal to the etsysPolicyProfileIndex, shall have the corresponding policy applied. Likewise, all stations corresponding to rows within the etsysStationPolicyProfileTable whose etsysStationPolicyProfileOperID is equal to the etsysPolicyProfileIndex, shall have the corresponding policy applied. The value of etsysPortPolicyProfileOperID for each such row in the etsysPortPolicyProfileTable will be equal to the etsysPortPolicyProfileAdminID, unless the authorization information from a source such as a RADIUS server indicates to the contrary.

Table/Group	Supported Variables	Comments
		Refer to the specific objects within this MIB as well as well as RFC2674, the CTRON-PRIORITY-CLASSIFY-MIB, the CTRON-VLAN-CLASSIFY-MIB, and the CTRON-RATE-POLICING-MIB for a complete explanation of the application and behavior of these objects. When this object's value is set to notInService(2) this policy will not be applied to any rows within the etsysPortPolicyProfileTable. To allow policy profiles to be applied for security implementations, setting this object's value from active(1) to notInService(2) or destroy(6) SHALL fail if one or more instances of etsysPortPolicyProfileOperID or etsysStationPolicyProfileOperID currently reference this entry's associated policy due to a set by an underlying security protocol such as RADIUS. For network functionality and clarity, setting this object to destroy(6) SHALL fail if one or more instances of etsysPortPolicyProfileOperID or etsysStationPolicyProfileOperID currently references this entry's etsysPolicyProfileIndex. Refer to the RowStatus convention for further details on the behavior of this object.
	etsysPolicyProfilePortVid Status	This object defines whether a PVID override should be applied to ports which have this profile active. enabled(1) means that any port with this policy active will have this row's etsysPolicyProfilePortVid applied to untagged frames or priority-tagged frames received on this port. disabled(2) means that etsysPolicyProfilePortVid will not be applied. When this object is set to disabled(2) the value of

Table/Group	Supported Variables	Comments
		etsysPolicyProfilePortVid has no meaning.
	etsysPolicyProfilePortVid	This object defines the PVID of this profile. If a port has an active policy and the policy's etsysPolicyProfilePortVidStatus is set to enabled(1), the etsysPolicyProfilePortVid will be applied to all untagged frames arriving on the port that do not match any of the policy classification rules. Note that the 802.1Q PVID will still exist from a management view but will NEVER be applied to traffic arriving on a port that has an active policy and enabled etsysPolicyProfilePortVid defined, since policy is applied to traffic arriving on the port prior to the assignment of a VLAN using the 802.1Q PVID. The behavior of an enabled etsysPolicyProfilePortVid on any associated port SHALL be identical to the behavior of the dot1qPvid upon that port. Note that two special, otherwise illegal, values of the etsysPolicyProfilePortVid are used in defining the default forwarding actions, to be used in conjunction with policy classification rules, and do not result in packet tagging: 0 Indicates that the default forwarding action is to drop all packets that do not match an explicit rule. 4095 Indicates that the default forwarding action is to forward any packets not matching any explicit rules.
	etsysPolicyProfilePriorityStatus	This object defines whether a Class of Service should be applied to ports which have this profile active. enabled(1) means that any port with this policy active will have etsysPolicyProfilePriority applied to this port.

Table/Group	Supported Variables	Comments
		disabled(2) means that etsysPolicyProfilePriority will not be applied. When this object is set to disabled(2) the value of etsysPolicyProfilePriority has no meaning.
	etsysPolicyProfilePriority	This object defines the default ingress Class of Service of this profile. If a port has an active policy and the policy's etsysPolicyProfilePriorityStatus is set to enabled(1), the etsysPolicyProfilePriority will be applied to all packets arriving on the port that do not match any of the policy classification rules. Note that dot1dPortDefaultUserPriority will still exist from a management view but will NEVER be applied to traffic arriving on a port that has an active policy and enabled etsysPolicyProfilePriority defined, since policy is applied to traffic arriving on the port prior to the assignment of a priority using dot1dPortDefaultUserPriority. The behavior of an enabled etsysPolicyProfilePriority on any associated port SHALL be identical to the behavior of the dot1dPortDefaultUserPriority upon that port.
	etsysPolicyProfileEgressVlans	The set of VLANs which are assigned by this policy to egress on ports for which this policy is active. Changes to a bit in this object affect the per-port per-VLAN Registrar control for Registration Fixed for the relevant GVRP state machine on each port for which this policy is active. A VLAN may not be added in this set if it is already a member of the set of VLANs in etsysPolicyProfileForbiddenVlans. This object is superseded on a per-port per-VLAN basis by any 'set' bits in dot1qVlanStaticEgressPorts and dot1qVlanForbiddenEgressPorts.

Table/Group	Supported Variables	Comments
		The default value of this object is a string of zeros.
	etsysPolicyProfileForbiddenVlans	The set of VLANs which are prohibited by this policy to egress on ports for which this policy is active. Changes to this object that cause a port to be included or excluded affect the per-port per-VLAN Registrar control for Registration Forbidden for the relevant GVRP state machine on each port for which this policy is active. A VLAN may not be added in this set if it is already a member of the set of VLANs in etsysPolicyProfileEgressVlans. This object is superseded on a per-port per-VLAN basis by any 'set' bits in the dot1qVlanStaticEgressPorts and dot1qVlanForbiddenEgressPorts. The default value of this object is a string of zeros.
	etsysPolicyProfileUntaggedVlans	The set of VLANs which should transmit egress packets as untagged on ports for which this policy is active. This object is superseded on a per-port per-VLAN basis by any 'set' bits in dot1qVlanStaticUntaggedPorts.
	etsysPolicyProfileOverwriteTCI	If set, the information contained within the TCI field of inbound, tagged packets will not be used by the device after the ingress classification stage of packet relay. The net effect will be that the TCI information may be used to classify the packet, but will be overwritten (and ignored) by subsequent stages of packet relay.
	etsysPolicyProfileRulePrecedence	Each octet will contain a single value representing the rule type to be matched against, defined by the PolicyClassificationRuleType textual convention. When read, will return the currently operating rule matching precedence, ordered from first consulted (in the first octet) to last consulted (in the last octet).

Table/Group	Supported Variables	Comments
		A set of a single octet of 0x00 will result in a reversion to the default precedence ordering. A set of any other values will result in the specified rule types being matched in the order specified, followed by the remaining rules, in default precedence order.
	etsysPolicyProfileVlanRFC3580Mappings	The set of VLANs which are currently being mapped onto this policy profile by the etsysPolicyRFC3580MapTable. This only refers to the mapping of vlan-tunnel-attributes returned from RADIUS in an RFC3580 context.
	etsysPolicyProfileMirrorIndex	A reference to a packet mirror destination (defined elsewhere). A value of (-1) indicates no mirror is specified, but a mirror is not explicitly prohibited. A value of (0) indicates that mirroring is explicitly prohibited, unless a higher precedence source (a rule) has specified a mirror.
	etsysPolicyProfileAuditSyslogEnable	Enables the sending of a syslog message if no rule bound to this profile has prohibited it.
	etsysPolicyProfileAuditTrapEnable	Enables the sending of a SNMP NOTIFICATION if no rule bound to this profile has prohibited it.
	etsysPolicyProfileDisablePort	Will set the ifOperStatus of the port, on which the frame which used this profile was received, to disable, if if no rule bound to this profile has prohibited it.
	etsysPolicyProfileUsageList	When read, a set bit indicates that this profile was used to send a syslog or trap message for corresponding port. When set, the native PortList will be bit-wise AND'ed with the set PortList, allowing the agent to clear the usage indication.
	etsysPolicyProfileFstIndex	A reference to a Flow Setup Throttling (FST) class as defined by the etsysFlowLimitingClassType object.

Table/Group	Supported Variables	Comments
		A value of (0) indicates no FST class is specified.
	etsysPolicyProfileHttpRedirectIndex	A reference to a HTTP Redirect server group as specified by the etsysPolicyHttpRedirectGroupIndex object. A value of (0) indicates no HTTP Redirect group is specified for this profile.
	etsysPolicyProfilePortAuthOverride	If a port has an active policy and that policy's etsysPolicyProfilePortAuthOverride is set to enabled(1), all frames arriving on the port will have that policy applied. In addition, any pre-existing entries with matching port values in the etsysMultiAuthSessionStationTable tables will change their authorization status to authTerminated(5). No further authentication will occur on this port. If disabled(2), the actions described above will not occur.

Table/Group	Supported Variables	Comments
etsysPolicyClassification group	etsysPolicyClassification MaxEntries	The maximum number of entries allowed in the etsysPolicyClassificationTable.
	etsysPolicyClassificationNumEntries	The current number of entries in the etsysPolicyClassificationTable.
	etsysPolicyClassificationLastChange	The sysUpTime at which the etsysPolicyClassificationTable was last modified.
	etsysPolicyClassificationTable	A table containing reference OIDs to entries within the classification tables. These classification tables include but may not be limited to: ctPriClassifyTable ctVlanClassifyTable ctRatePolicyingConfigTable This table is used to map a list of classification rules to an instance of the etsysPolicyProfileTable.
	etsysPolicyClassificationEntry	Describes a particular entry within the etsysPolicyClassificationTable. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPolicyClassificationIndex	Administratively assigned unique value, greater than zero. Each etsysPolicyClassificationIndex instance MUST be unique within the scope of its associated etsysPolicyProfileIndex.
	etsysPolicyClassificationOID	This object follows the RowPointer textual convention and is an OID reference to a classification rule. This object MUST NOT be modifiable while this entry's etsysPolicyClassificationStatus object has a value of active(1).
	etsysPolicyClassificationRowStatus	The status of this row. When set to active(1) this entry's classification rule, as referenced by etsysPolicyClassificationOID, becomes one of its associated policy's set of rules.

Table/Group	Supported Variables	Comments
		When this entry's associated policy, as defined by etsysPolicyProfileIndex, is active and assigned to a port through the etsysPortPolicyProfileTable or to a station through the etsysStationPolicyProfileTabbe, this classification rule will be applied to the port or station. The exact behavior of this application depends upon the classification rule. When this object is set to notInService(2) or notReady(3) this entry is not considered one of its associated policy's set of rules and this classification rule will not be applied. An entry MAY NOT be set to active(1) unless this row's etsysPolicyClassificationOID is set to a valid classification rule.
	etsysPolicyClassificationIngressList	The ports on which an active policy profile has defined this classification rule applies.
etsysPortPolicyProfile group	etsysPortPolicyProfileLastChange	sysUpTime at which the etsysPortPolicyProfileTable was last modified.
	etsysPortPolicyProfileEntry	Describes a particular entry within the etsysPortPolicyProfileTable. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPortPolicyProfileIndexType	This object defines the specific type of port this entry represents.
	etsysPortPolicyProfileIndex	An index value which represents a unique port of the type defined by this entry's etsysPortPolicyProfileIndexType.

Table/Group	Supported Variables	Comments
	etsysPortPolicyProfileAdminID	This object represents the desired Policy Profile for this dot1dBasePort or this ifIndex. Setting this object to any value besides zero (0) should, if possible, immediately place this entry's dot1dBasePort or ifIndex into the given Policy Profile. This object and etsysPortPolicyProfileOperID may not be the same if this object is set to a Policy (i.e. an instance of the etsysPolicyProfileTable) which is not in an active state or if the etsysPortPolicyProfileOperID has been set by an underlying security protocol such as RADIUS.
	etsysPortPolicyProfileOperID	This object is the current policy which is being applied to this entry's dot1dBasePort. A value of zero(0) indicates there is no policy being applied to this dot1dBasePort or this ifIndex. If the value of this object has been set by an underlying security protocol such as RADIUS, sets to this entry's etsysPortPolicyProfileAdminID MUST NOT change the value of this object until such time as the security protocol releases this object by setting it to a value of zero (0).
	etsysPortPolicyProfileSummaryTable	his table provides aggregate port information on a per policy, per port type basis.
	etsysPortPolicyProfileSummaryEntry	Conceptually defines a particular entry within the etsysPortPolicyProfileSummaryTable.
	EtsysPortPolicyProfileSummaryEntry	This object defines the specific type of port this entry represents.
	etsysPortPolicyProfileSummaryAdminID	An aggregate list of all Ports currently supporting rules which assign this profileIndex through administrative means. Rules of this type have a valid etsysPolicyRuleResult2 action and a profileIndex of 0.

Table/Group	Supported Variables	Comments
	etsysPortPolicyProfileSummaryOperID	An aggregate list of all Ports currently supporting rules which assign this profileIndex through either an administrative or dynamic means. The profileId which will be assigned operationally, as frames are handled are too be reported here.
	etsysPortPolicyProfileSummaryDynamicID	An aggregate list of all Ports currently supporting rules which assign this profileIndex through a dynamic means. For example the profileIndex returned via a successful 802.1X supplicant authentication.

Table/Group	Supported Variables	Comments
etsysStationPolicyProfile group	etsysStationPolicyProfileMaxEntries	The maximum number of entries allowed in the etsysStationPolicyProfileTable. If this number is exceeded, based on stations connecting to the edge device, the oldest entries will be deleted.
	etsysStationPolicyProfileNumEntries	The current number of entries in the etsysStationPolicyProfileTable.
	etsysStationPolicyProfileLastChange	sysUpTime at which the etsysStationPolicyProfileTable was last modified.
	etsysStationPolicyProfileTable	This table allows for a one to one mapping between a station's identifying address and a Policy Profile.
	etsysStationPolicyProfileEntry	Describes a particular entry within the etsysStationPolicyProfileTable. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysStationPolicyProfileIndex	An index value which represents a unique station entry.
	etsysStationIdentifierType	Indicates the type of station identifying address contained in etsysStationIdentifier.
	etsysStationIdentifier	A value which represents a unique MAC Address, IP Address, or other identifying address for a station, or other logical and authenticatable sub-entity within a station, connected to a port.
	etsysStationPolicyProfileOperID	This object is the current policy which is being applied to this entry's MAC Address. A value of zero(0) indicates there is no policy being applied to this MAC Address. The value of this object reflects either the setting from an underlying AAA service such as RADIUS, or the default setting based on the etsysPortPolicyProfileAdminID for the port on which the station is connected.

Table/Group	Supported Variables	Comments
		This object and the corresponding etsysPortPolicyProfileAdminID will not be the same if this object has been set by an underlying security protocol such as RADIUS.
	etsysStationPolicyProfilePortType	A textual convention that defines the specific type of port designator the corresponding entry represents.
	etsysStationPolicyProfilePortID	A value which represents the physical port, of the type defined by this entry's etsysStationPolicyProfilePortType, on which the associated station entity is connected. This object is for convenience in cross referencing stations to ports.
etsysInvalidPolicyPolicy group	etsysInvalidPolicyAction	Specifies the action that the edge device should take if asked to apply an invalid or unknown policy. applyDefaultPolicy(1) - Ignore the result and search for the next policy assignment rule. dropPackets(2) - Block traffic. forwardPackets(3) - Forward traffic, as if no policy had been assigned (via 802.1D/Q rules). Although dropPackets(2) is the most secure option, it may not always be desirable.
	etsysInvalidPolicyCount	Increments to indicate the number of times the device has detected an invalid/unknown policy.
etsysDevicePolicyProfile group	etsysDevicePolicyProfileDefault	If this value is non-zero, the value indicates the etsysPolicyProfileEntry (and its associated etsysPolicyClassificationTable entries) which should be used by the device if the device is incapable of using the profile (or specific parts of the profile) explicitly applied to an inbound frame. A value of zero indicates that no default profile is currently active.

Table/Group	Supported Variables	Comments
etsysPolicyCapability group	etsysPolicyCapabilities	A list of capabilities related to policies. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyDynaPIDRuleCapabilities	A list of rule types which are supported by this device for the purpose of dynamically assigning a profile to the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyAdminPIDRuleCapabilities	A list of rule types which are supported by this device for the purpose of administratively assigning a profile to the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyVlanRuleCapabilities	A list of rule types which are supported by this device for the purpose of assigning a VlanId to the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyCosRuleCapabilities	A list of rule types which are supported by this device for the purpose of assigning a CoS to the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyDropRuleCapabilities	A list of rule types which are supported by this device for the purpose of discarding the

Table/Group	Supported Variables	Comments
		network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyForwardRuleCapabilities	A list of rule types which are supported by this device for the purpose of forwarding the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicySyslogRuleCapabilities	A list of rule types which are supported by this device for the purpose of issuing syslog messages when the rule is used to identify the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyTrapRuleCapabilities	A list of rule types which are supported by this device for the purpose of issuing an SNMP notify (trap) messages when the rule is used to identify the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyDisablePortRuleCapabilities	A list of rule types which are supported by this device for the purpose of disabling the ingress port identified when the rule matches the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates

Table/Group	Supported Variables	Comments
		the described functionality is not supported.
	etsysPolicySupportedPortList	The list ports which support policy profile assignment (i.e. the ports which _do_ policy). This object may be useful to management entities which desire to scope action to only those ports which support policy. A port which appears in this list, must support, at minimum, the assignment of a policy profile to all traffic ingressing the port.
	etsysPolicyEnabledTable	This table allows for the configuration of policy profile assignment methods, per port, including the ability to disable policy profile assignment, per port. In addition, a ports capabilities, with respect to policy profile assignment are reported.
	etsysPolicyEnabledTableEntry	Describes a particular entry within the etsysPolicyEnabledTable.
	etsysPolicyEnabledSupportedRuleTypes	The list of rule types which the devices supports for the purpose of assigning policy profiles to network traffic ingressing this dot1dBasePort.
	etsysPolicyEnabledEnabledRuleTypes	The list of rule types from which the device will assign policy profiles to network traffic ingressing this dot1dBasePort. Rules which have a type not enumerated here must not be used to assign policy profiles, but must still be used to interrogate the rule-set bound to the determined policy profile. A set of all cleared bits will effectively disable policy in the port.
	etsysPolicyEnabledEgressEnabled	Controls the enabling and disabling the application of policy as packets egress the switching process on the dot1dBasePort specified in the indexing.
	etsysPolicyRuleAttributeTable	This table details each supported rule type attribute

Table/Group	Supported Variables	Comments
		for rule data length in bytes, rule data length in bits, and the maximum number of rules that may use that type.
	etsysPolicyRuleAttributeTableEntry	Describes a particular entry within the etsysPolicyRuleAttributeTable.
	etsysPolicyRuleAttributeByteLength	This rule type's maximum length, in bytes of the etsysPolicyRuleData. Devices supporting this object MUST allow sets for this rule data of any valid length up to and including the length value represented by this object. Management entities must also expect to read back the maximum data length for each type regardless of the length the data was set with.
	etsysPolicyRuleAttributeBitLength	This rule type's maximum bit length for traffic data. This value also represents the maximum mask that may be used for rule data. The mask MUST NOT exceed the rule data size. Masks that exceed the data size shall be considered invalid and result in an SNMP set failure.
	etsysPolicyRuleAttributeMaxCreatable	If this value is non-zero, the value indicates the maximum number of rules of this type the agent can support.
	etsysPolicyRuleTciOverwriteCapabilities	A list of rule types which are supported by this device for the purpose of overwriting the TCI in received packets described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates the described functionality is not supported.
	etsysPolicyRuleMirrorCapabilities	A list of rule types which are supported by this device for the purpose of mirroring the network traffic described by the bit. A set bit, with the value 1, indicates support for the described functionality. A clear bit, with the value 0, indicates

Table/Group	Supported Variables	Comments
		the described functionality is not supported.
	etsysPolicyRuleQuarantineCapabilities	A list of rule types which are supported by this device for the purpose of quarantining the user to a specified profile id when the rule is used to identify the network traffic described by the bit. A set bit, with the value of 1, indicates support for the described functionality. A clear bit, with the value of 0, indicates the described functionality is not supported.
etsysPolicyMap group	etsysPolicyMapMaxEntries	This has been obsoleted.
	etsysPolicyMapNumEntries	This has been obsoleted.
	etsysPolicyMapLastChange	This has been obsoleted.
	etsysPolicyMapPvidOverRide	This has been obsoleted.
	etsysPolicyMapUnknownPvidPolicy	This has been obsoleted.
	etsysPolicyMapTable	This has been obsoleted.
	etsysPolicyMapEntry	This has been obsoleted.
	etsysPolicyMapIndex	This has been obsoleted.
	etsysPolicyMapRowStatus	This has been obsoleted.
	etsysPolicyMapStartVid	This has been obsoleted.
	etsysPolicyMapEndVid	This has been obsoleted.
	etsysPolicyMapPolicyIndex	This has been obsoleted.

Table/Group	Supported Variables	Comments
etsysPolicyRules group	etsysPolicyRulesMaxEntries	The maximum number of entries allowed in the etsysPolicyRulesTable.
	etsysPolicyRulesNumEntries	The current number of entries in the etsysPolicyRulesTable.
	etsysPolicyRulesLastChange	The sysUpTime at which the etsysPolicyRulesTable was last modified.
	etsysPolicyRulesAccountingEnable	Controls the collection of rule usage statistics. If disabled, no usage statistics are gathered and no auditing messages will be sent. When enabled, rule will gather usage statistics, and auditing messages will be sent, if enabled for a given rule.
	etsysPolicyRulesPortDisabledList	A portlist containing bits representing the dot1dBridgePorts which have been disabled via the mechanism described in the etsysPolicyRuleDisablePort leaf. A set bit indicates a disabled port. Ports may be enabled by performing a set with the corresponding bit cleared. Bits which are set will be ignored during the set operation.

Table/Group	Supported Variables	Comments
etsysPolicyRuleTable	etsysPolicyRuleTable	A table containing rules bound to individual policies. A Rule is comprised of three components, a unique description of the network traffic, an associated list of actions, and an associated list of accounting and auditing controls and information. The unique description of the network traffic, defined by a PolicyClassificationRuleType together with a length, matching data and a relevant bits field, port type, and port number (port number zero is reserved to mean any port), and scoped by a etsysPolicyProfileIndex, is used as the table index.
	etsysPolicyRuleEntry	Describes a particular entry within the etsysPolicyRuleTable. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPolicyRuleProfileIndex	The etsysPolicyProfileIndex for which the rule is defined. A value of zero(0) has special meaning in that it scopes rules which are used to determine the Policy Profile to which the frame belongs. See the etsysPolicyRuleResult1 and etsysPolicyRuleResult2 descriptions for specifics of how the results of a rule hit differ when the etsysPolicyRuleProfileIndex is zero.
	etsysPolicyRuleType	The type of network traffic reference by the etsysPolicyRuleData.
	etsysPolicyRuleData	The data pattern to match against, as defined by the etsysPolicyRuleType, encoded in network-byte order.
	etsysPolicyRulePrefixBits	The relevant number of bits defined by the etsysPolicyRuleData, to be used when matching against a frame, relevant bits are specified in

Table/Group	Supported Variables	Comments
		longest-prefix-first style (left to right). A value of zero carries the special meaning of all bits are relevant.
	etsysPolicyRulePortType	The port number on which the rule will be applied. Zero(0) is a special case, indicating that the rule should be applied to all ports.
	etsysPolicyRulePort	The port number on which the rule will be applied. Zero(0) is a special case, indicating that the rule should be applied to all ports.
	etsysPolicyRuleRowStatus	The status of this row. When set to active(1) this entry's classification rule, as referenced by etsysPolicyRulesOID, becomes one of its associated policy's set of rules. When this entry's associated policy, as defined by etsysPolicyRuleProfileIndex, is active and assigned to a port through the etsysPortPolicyProfileTable or to a station through the etsysStationPolicyProfileTabbe, this classification rule will be applied to the port or station. The exact behavior of this application depends upon the classification rule. When this object is set to notInService(2) or notReady(3) this entry is not considered one of its associated policy's set of rules and this classification rule will not be applied.
	etsysPolicyRuleStorageType	The storage type of this row. When set to volatile(1) this entry's classification rule, as referenced by etsysPolicyRulesOID, will be removed (if present) from non-volatile storage. Rows created dynamically by the device will typically report this as their default storage type. When set to nonVolatile(1) this entry's classification

Table/Group	Supported Variables	Comments
		rule, as referenced by etsysPolicyRulesOID, will be added to non- volatile storage. This is the default value for rows created as the result of external management. Values of other(0), permanent(4), and readOnly(5) may not be set, although they may be returned for rows created by the device.
	etsysPolicyRuleUsageList	When read, a set bit indicates that this rule was used to classify traffic on the corresponding port. When set, the native PortList will be bit-wise AND'ed with the set PortList, allowing the agent to clear the usage indication.
	etsysPolicyRuleResult1	If the etsysPolicyRuleProfileIndex is 0 then this field is read-only and defines the profile ID which will be assigned to frames matching this rule. This is the dynamically assigned value and may differ from the administratively configured value. If the etsysPolicyRuleProfileIndex is not 0 then this field is read-create and defines the VLAN ID with which to mark a frame matching this PolicyRule. Note that three special, otherwise illegal, values of the etsysPolicyRuleVlan are used in defining the forwarding action. -1 Indicates that no VLAN or forwarding behavior modification is desired. A rule will not be matched against for the purpose of determining a marking VID if this value is set. 0 Indicates that the default forwarding action is to drop the packets matching this rule. 4095 Indicates that the default forwarding action is to forward any packets matching this rule.
	etsysPolicyRuleResult2	If the etsysPolicyRuleProfileIndex is 0

Table/Group	Supported Variables	Comments
		then this field is read-create and defines the profile ID which the managing entity desires assigned to frames matching this rule. This is the administrative value and may differ from the dynamically assigned active value. If the etsysPolicyRuleProfileIndex is not 0 then this field is The CoS with which to mark a frame matching this PolicyRule. Note that one special, otherwise illegal, values of the etsysPolicyRuleCoS are used in defining the forwarding action. -1 Indicates that no CoS or forwarding behavior modification is desired. A rule will not be matched against for the purpose of determining a CoS if this value is set.
	etsysPolicyRuleAuditSyslogEnable	Controls the sending of a syslog message when a bit in the etsysPolicyRuleUsageList transitions from 0 to 1.
	etsysPolicyRuleAuditTrapEnable	Controls the sending of an SNMP NOTIFICATION when a bit in the etsysPolicyRuleUsageList transitions from 0 to 1.
	etsysPolicyRuleDisablePort	Controls the disabling of a port (ifOperStatus of the corresponding ifIndex will be down) when a bit in the etsysPolicyRuleUsageList transitions from 0 to 1. When set to enabled, the corresponding ifIndex will be disabled upon the transition.
	etsysPolicyRuleOperPid	If the etsysPolicyRuleProfileIndex is 0 then this field contains the currently applied profile ID for frames matching this rule. This may be either the administratively applied value or the dynamically applied value. If the etsysPolicyRuleProfileIndex is not 0, then this object will return -1.

Table/Group	Supported Variables	Comments
		Note that one special value exists: -1 Indicates that no profile ID is being applied by this rule.
	etsysPolicyRuleOverwriteTCI	If set, the information contained within the TCI field of inbound, tagged packets will not be used by the device after the ingress classification stage of packet relay. The net effect will be that the TCI information may be used to classify the packet, but will be overwritten (and ignored) by subsequent stages of packet relay.
	etsysPolicyRuleMirrorIndex	A reference to a packet mirror destination (defined elsewhere). A value of (-1) indicates no mirror is specified, but a mirror is not explicitly prohibited. A value of (0) indicates that mirroring is explicitly prohibited, unless a higher precedence rule has specified a mirror.
	etsysPolicyRuleQuarantineProfileIndex	If the etsysPolicyRuleProfileIndex is not 0 this field defines the profile ID which will be used as the quarantine provisioning agents mux response for the mac address and port whose frames matched this rule. A value of (-1) indicates no quarantine profile is specified, but quarantine is not explicitly prohibited. A value of (0) indicates that quarantine is explicitly prohibited. If the etsysPolicyRuleProfileIndex is 0 this field is read only and will always return -1 when read.
	etsysPolicyRuleHttpRedirectIndex	A reference to a HTTP Redirect server group as specified by the etsysPolicyHttpRedirectGroupIndex object. A value of (-1) indicates no HTTP Redirect is specified, but HTTP redirection is not explicitly prohibited.

Table/Group	Supported Variables	Comments
		A value of (0) indicates that HTTP Redirect is explicitly prohibited, unless a higher precedence rule has specified a HTTP Redirect. Packets are only subject to HTTP redirection if they are IP frames with TCP port numbers matching an entry in etsysPolicyHttpRedirectSocketTable.

Table/Group	Supported Variables	Comments
etsysPolicyRulePortTable	etsysPolicyRulePortTable	The purpose of this table is to provide an agent the ability to easily determine which rules have been used on a given bridge port. A row will only be present when the rule which the instancing describes has been used. The agent may remove a row (and clear the used status) by setting the etsysPolicyRulePortHit leaf to False. PolicyClassificationRuleType together with a length, matching data and a relevant bits field, port type, and port number (port number zero is reserved to mean any port), scoped by a etsysPolicyRuleProfileIndex, and preceded by a dot1dBasePort is used as the table index.
	etsysPolicyRulePortEntry	
	etsysPolicyRulePortHit	Every row will report a value of True, indicating that the Rule described by the instancing was used on the given port. An agent may be set this leaf to False to clear remove the row and clear the Rule Use bit for the specified Rule, on the given bridgePort.
	etsysPolicyRuleDynamicProfileAssignmentOverride	If true, administratively assigned profile assignment rules override dynamically assigned profiles assignments for a given rule. If false, the dynamically assigned value (typically created by a successful authentication attempt) overrides the administratively configured value. The agent may optionally implement this leaf as read-only.
	etsysPolicyRuleDefaultDynamicSyslogStatus	If enabled(1), rules dynamically created will set etsysPolicyRuleAuditSyslogEnable to enabled. If disabled(2) a dynamically created rule will have etsysPolicyRuleAuditSyslogEnable set to disabled. The agent may optionally implement this leaf as read-only.
	etsysPolicyRuleDefaultDy	If enabled(1), rules

Table/Group	Supported Variables	Comments
	namicTrapStatus	dynamically created will set etsysPolicyRuleAuditTrapEnable to enabled. If disabled(2) a dynamically created rule will have etsysPolicyRuleAuditTrapEnable set to disabled. The agent may optionally implement this leaf as read-only.
	etsysPolicyRuleStatsAutoClearOnLink	If set to enabled(1), when operstatus up is detected on any port the agent will clear the rule usage information associated with that port. This ability is further scoped to the list of ports defined by etsysPolicyRuleStatsAutoClearPorts. This leaf is optional and will have no effect on an agent which has rule use accounting disabled or does not support rule use accounting. By default, the rule use accounting information will not be modified by operstatus transitions.
	etsysPolicyRuleStatsAutoClearInterval	The interval at which the device will automatically clear rule usage statistics, in minutes. This ability is disabled (usage statistics will not be automatically cleared) if set to zero(0). This ability is further scoped to the list of ports defined by etsysPolicyRuleStatsAutoClearPorts. This leaf is optional and will have no effect on an agent which has rule use accounting disabled or does not support rule use accounting.
	etsysPolicyRuleStatsAutoClearPorts	The list ports on which rule usage statistics will be cleared by one of the AutoClear actions (etsysPolicyRuleStatsAutoClearInterval, etsysPolicyRuleStatsAutoClearOnProfile, or etsysPolicyRuleStatsAutoClearOnLink). By default, no ports will be set in this list. This leaf is optional, unless the agent claims support for one of the other 'autoclear' objects, and will have no effect on an agent

Table/Group	Supported Variables	Comments
		which has rule use accounting disabled or does not support rule use accounting.
	etsysPolicyRuleStatsAutoClearOnProfile	If set to enabled(1), when a rule assigning a PolicyProfile (whose etsysPolicyRuleProfileIndex is zero(0)) is activated, all the rule usage bits associated with the rules bound to the PolicyProfile specified by the etsysPolicyRuleOperPid and the port specified by the etsysPolicyRulePort are cleared (if there is no port specified or no valid etsysPolicyRuleProfileIndex specified, then no action follows). This ability is further scoped to the list of ports defined by etsysPolicyRuleStatsAutoClearPorts. This leaf is optional and will have no effect on an agent which has rule use accounting disabled or does not support rule use accounting. By default, the rule use accounting information will not be modified by the creation or activation of PolicyProfile assignment rules." DEFVAL { disabled }
	etsysPolicyRuleStatsDroppedNotifications	A count of the number of times the agent has dropped notification (syslog or trap) of a etsysPolicyRuleUsageList bit transition. A management entity might use this leaf as an indication to read the etsysPolicyRuleUsageList objects for important rules. This count should be kept to the best of the device's ability, and explicitly does not cover notifications discarded by the network.
	etsysPolicyRuleSylogMachineReadableFormat	If enabled, the device should format rule usage messages so that they might be processed by a machine (scripting backend, etc). If disabled, the messages

Table/Group	Supported Variables	Comments
		should be formatted for human consumption.
	etsysPolicyRuleSylogExtendedFormat	If enabled, the device should provide additional information in rule-hit syslog messages. This information MAY include what actions may have been initiated by the rule (if any) or data mined from the packet which matched the rule.
	etsysPolicyRuleSylogEveryTime	If enabled, the device will syslog on every rule hit (or profile hit) which specifies SYSLOG as the action, instead of only when the associated bit in the etsysPolicyProfileUsageList or the etsysPolicyRuleUsageList is clear. It should be noted that this may cause MANY messages to be generated.

Table/Group	Supported Variables	Comments
etsysPolicyNonVolatileRuleTable	etsysPolicyNonVolatileRulesLastChange	The sysUpTime at which the etsysPolicyNonVolatileRuleTable was last modified.
	etsysPolicyNonVolatileRuleTable	A table containing rules bound to individual policies. The rules here contained are representations of the non-volatile rules contained in the etsysPolicyRuleTable defined in this MIB. A Rule comprises three components, a unique description of the network traffic, an associated list of actions, and an associated list of accounting and auditing controls and information. The unique description of the network traffic, defined by a PolicyClassificationRuleType together with a length, matching data and a relevant bits field, port type, and port number (port number zero is reserved to mean any port), and scoped by a etsysPolicyProfileIndex, is used as the table index.
	etsysPolicyNonVolatileRuleEntry	Describes a particular entry within the etsysPolicyNonVolatileRuleTable. Entries within this table are representations of the non-volatile entries found in the etsysPolicyRuleTable.
	etsysPolicyNonVolatileRuleRowStatus	The status of this row. When active(1) this entry's classification rule, is one of its associated policy's set of rules. When this entry's associated policy, as defined by etsysPolicyRuleProfileIndex, is active and assigned to a port through the etsysPortPolicyProfileTable or to a station through the etsysStationPolicyProfileTable, this classification rule will be applied to the port or station. The exact behavior of this application depends upon the classification rule.

Table/Group	Supported Variables	Comments
		When this object is notInService(2) or notReady(3) this entry is not considered one of its associated policy's set of rules and this classification rule will not be applied.
	etsysPolicyNonVolatileRuleStorageType	The storage type of this row. For all entries in the table this is set to nonVolatile(1), and as such, this entry's classification rule will be added to non-volatile storage.
	etsysPolicyNonVolatileRuleUsageList	When read, a set bit indicates that this rule was used to classify traffic on the corresponding port.
	etsysPolicyNonVolatileRuleResult1	If the etsysPolicyRuleProfileIndex is 0 this field defines the profile ID which will be assigned to frames matching this rule. This is the dynamically assigned value and may differ from the administratively configured value. If the etsysPolicyRuleProfileIndex is not 0 then this field defines the VLAN ID with which to mark a frame matching this PolicyNonVolatileRule. Note that three special, otherwise illegal, values of the etsysPolicyNonVolatileRuleVlan are used in defining the forwarding action. -1 Indicates that no VLAN or forwarding behavior modification is desired. A rule will not be matched against for the purpose of determining a marking VID if this value is set. 0 Indicates that the default forwarding action is to drop the packets matching this rule. 4095 Indicates that the default forwarding action is to forward any packets matching this rule.
	etsysPolicyNonVolatileRuleResult2	If the etsysPolicyRuleProfileIndex is 0 this field defines the profile

Table/Group	Supported Variables	Comments
		ID which the managing entity desires assigned to frames matching this rule. This is the administrative value and may differ from the dynamically assigned active value. If the etsysPolicyRuleProfileIndex is not 0 then this field is The CoS with which to mark a frame matching this PolicyNonVolatileRule. Note that one special, otherwise illegal, value of the etsysPolicyNonVolatileRuleCoS is used in defining the forwarding action. -1 Indicates that no CoS or forwarding behavior modification is desired. A rule will not be matched against for the purpose of determining a CoS if this value is set.
	etsysPolicyNonVolatileRuleAuditSyslogEnable	If enabled, a syslog message is sent when a bit in the etsysPolicyNonVolatileRuleUsageList transitions from 0 to 1.
	etsysPolicyNonVolatileRuleAuditTrapEnable	If enabled, an SNMP NOTIFICATION is sent when a bit in the etsysPolicyNonVolatileRuleUsageList transitions from 0 to 1.
	etsysPolicyNonVolatileRuleDisablePort	If enabled, a port is disabled (ifOperStatus of the corresponding ifIndex will be down) when a bit in the etsysPolicyNonVolatileRuleUsageList transitions from 0 to 1.
	etsysPolicyNonVolatileRuleOperPid	If the etsysPolicyRuleProfileIndex is 0 then this field contains the currently applied profile ID for frames matching this rule. This may be either the administratively applied value or the dynamically applied value. If the etsysPolicyRuleProfileIndex is not 0, then this object will return -1. Note that one special value exists: -

Table/Group	Supported Variables	Comments
		1 Indicates that no profile ID is being applied by this rule.
	etsysPolicyNonVolatileRuleOverwriteTCI	If set, the information contained within the TCI field of inbound, tagged packets will not be used by the device after the ingress classification stage of packet relay. The net effect will be that the TCI information may be used to classify the packet, but will be overwritten (and ignored) by subsequent stages of packet relay.
	etsysPolicyNonVolatileRuleMirrorIndex	A reference to a packet mirror destination (defined elsewhere). A value of (-1) indicates no mirror is specified, but a mirror is not explicitly prohibited. A value of (0) indicates that mirroring is explicitly prohibited, unless a higher precedence rule has specified a mirror.
	etsysPolicyNonVolatileRuleQuarantineProfileIndex	If the etsysPolicyRuleProfileIndex is not 0 this field defines the profile ID which will be used as the quarantine provisioning agents mux response for the mac address and port whose frames matched this rule. A value of (-1) indicates no quarantine profile is specified, but quarantine is not explicitly prohibited. A value of (0) indicates that quarantine is explicitly prohibited. If the etsysPolicyRuleProfileIndex is 0 this field will always return -1 when read.

Table/Group	Supported Variables	Comments
etsysPolicyRFC3580Map group	etsysPolicyRFC3580MapResolveReponseConflict	Indicates which field to use in the application of the RADIUS response in the event that both the proprietary filter-id indicating a policy profile and the standard (RFC3580) vlan-tunnel-attribute are present. If policyProfile(1) is selected, then the filter-id will be used, if vlanTunnelAttribute(2) is selected, then the vlan-tunnel-attribute will be used (and the policy-map will be applied, if present). A value of vlanTunnelAttributeWithPolicyProfile(3) indicates that both attributes should be applied, in the following manner: the policyProfile should be enforced, with the exception of the etsysPolicyProfilePortVid (if present), the returned vlan-tunnel-attribute will be used in its place. In this case, the policy-map will be ignored (as the policyProfile was explicitly assigned). VLAN classification rules will still be applied, as defined by the assigned policyProfile. Modifications of this value will not effect the current status of any users currently authenticated. The new state will be applied to new, successful authentications. The current status of current authentication may be modified through the individual agents or through the ENTERASYS-MULTI-AUTH-MIB, if supported.
	etsysPolicyRFC3580MapLastChange	The value of sysUpTime when the etsysPolicyRFC3580MapTable was last modified.
	etsysPolicyRFC3580MapTableDefault	If read as True, then the etsysPolicyRFC3580MapTable is in the default state (no mappings have been created), if False, then non-default mappings exist. If set to True, then the etsysPolicyRFC3580MapTable will be put into the default state

Table/Group	Supported Variables	Comments
		(no mappings will exist). A set to False is not valid and MUST fail.
	etsysPolicyRFC3580MapTable	A table containing VLAN ID to policy mappings. A policy is a group of classification rules which may be applied on a per user basis, to ports or to stations.
	etsysPolicyRFC3580MapEntry	Conceptually defines a particular entry within the etsysPolicyRFC3580MapTable. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPolicyRFC3580MapVlanId	The VlanIndex which will map to the policy profile specified by the etsysPolicyRFC3580MapPolicyIndex of this row. This will be used to map the VLAN returned by value from the Tunnel- Private-Group-ID RADIUS attribute.
	etsysPolicyRFC3580MapPolicyIndex	The index of a Policy Profle as defined in the etsysPolicyProfileTable. A value of 0 indicates that the row is functionally non- operational (no mapping exists). Devices which support the ENTERASYS-VLAN-AUTHORIZATION-MIB, and for which the value of etsysVlanAuthorizationEnable is Enabled and the value of etsysVlanAuthorizationStatus is Enabled on the port referenced by the authorization request, should then use the VlanIndex provisioned (e.g. from the Tunnel-Private-Group-ID RADIUS attribute) as defined by RFC3580, otherwise, the device should treat the result as if no matching Policy Profile had been found (e.g. as a simple success). In the case where a Policy Profile is already being applied to the referenced station, but no mapping exists, the device MUST treat the Tunnel-Private-Group-ID as an override to the etsysPolicyProfilePortVid defined by that profile (any

Table/Group	Supported Variables	Comments
		matched classification rules which explicit provision a VLAN MUST still override both the etsysPolicyProfilePortVid and the Tunnel-Private-Group-ID.) A non-zero value of this object indicates that the VlanIndex provisioned (e.g. from the Tunnel-Private-Group-ID RADIUS attribute) should be mapped to a Policy Profile as defined in the etsysPolicyProfileTable, and that policy applied as if the Policy name had been provisioned instead (e.g, in the Filter-ID RADIUS attribute). If the mapping references a non-existent row of the etsysPolicyProfileTable, or the referenced row has a etsysPolicyProfileRowStatus value other than Active, the device MUST behave as if the mapping did not exist (apply the vlan-tunnel-attribute). The etsysPolicyRFC3580MapInvalid Mapping MUST then be incremented.
	etsysPolicyRFC3580MapInvalidMapping	Increments to indicate the number of times the device has detected an invalid/unknown EtsysPolicyRFC3580MapEntry (i.e. one that references an in-active or non-existent etsysPolicyProfile).

Table/Group	Supported Variables	Comments
etsysPolicyHttpRedirect group	etsysPolicyHttpRedirectMaxNumSockets	The maximum number of TCP port numbers the device may listen on simultaneously for HTTP redirection.
	etsysPolicyHttpRedirectMaxNumServer	The number of servers that may be configured per server group in the etsysPolicyHttpRedirectServerTable.
	etsysPolicyHttpRedirectSocketTable	A table containing TCP sockets the device will listen on for HTTP traffic to redirect. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPolicyHttpRedirectSocketEntry	Conceptually specifies a TCP socket the device will listen on for HTTP traffic to redirect.
	etsysPolicyHttpRedirectSocketIndex	An arbitrary index from 1 to etsysPolicyHttpRedirectMaxNumSockets.
	etsysPolicyHttpRedirectListenSocket	TCP port number (1-65535) that the device will listen on for HTTP traffic suitable for redirection. A value of 0 indicates that this entry does not specify a TCP socket to listen on.
	etsysPolicyHttpRedirectServerTable	A table containing HTTP redirect server group entries.
	etsysPolicyHttpRedirectServerEntry	Conceptually defines a HTTP redirect server group. Within each group, one or more redirect servers may be defined. HTTP redirects will be sent to different servers within a group using a round-robin algorithm. Entries within this table MUST be considered non-volatile and MUST be maintained across entity resets.
	etsysPolicyHttpRedirectGroupIndex	The index referenced by etsysPolicyProfileHttpRedirectIndex to identify a particular HTTP redirect server group. The maximum value of this index is specified by the etsysPolicyHttpRedirectNumServerGroups object.
	etsysPolicyHttpRedirectServerIndex	The index for a particular server within the redirect group. The

Table/Group	Supported Variables	Comments
		maximum value of this index is specified by the etsysPolicyHttpRedirectMaxNumServer object.
	etsysPolicyHttpRedirectServerUri	The absolute URI on the redirect server to redirect the user to. This object MUST specify the scheme, authority and path. The URI may optionally include a query and/or fragment portions as well.
	etsysPolicyHttpRedirectServerStatus	A value of enabled(1) causes the entry to be made ready for use in redirecting HTTP traffic. A set of enabled(1) will only succeed if the the other entries with STATUS of read-write in table have been set to appropriate non-default values.
etsysPolicySystem group	etsysPolicyEnabledState	Controls the enabling and disabling the entire Policy application. A value of enabled(1) indicates that all objects in this MIB module are actively being applied on the device. A value of disabled(2) indicates that none of the objects in this MIB are actively being applied. The agent may optionally implement this leaf as read-only. All other objects in this MIB module MUST remain available and configurable regardless of the current value of this object. This object MUST be considered non-volatile and its value MUST be maintained across entity resets.

ENTERASYS-RADIUS-ACCT-CLIENT-EXT-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
MIB Objects	etsysRadiusAcctClientEnable	This object indicates whether or not RADIUS Accounting is enabled or disabled. This parameter value is maintained across system reboots.
	etsysRadiusAcctClientUpdateInterval	This indicates how many seconds elapse between accounting interim updates. This parameter value is maintained across system reboots. A value of zero means no Interim Updates. If the value is less than etsysRadiusAcctClientIntervalMinimum, the etsysRadiusAcctClientIntervalMinimum value will be used for the update interval time. If RADIUS Accounting is not enabled, this object is ignored. Note that Accounting Interim Updates are not issued by the RADIUS Accounting Client, unless so requested by the RADIUS Server in an Access Accept packet.
	etsysRadiusAcctClientIntervalMinimum	This indicates the minimum value in seconds between accounting interim updates supported by the managed entity. This parameter value is maintained across system reboots. If RADIUS Accounting is not enabled, this object is ignored.
	etsysRadiusAcctClientServerTable	The (conceptual) table listing the RADIUS Accounting servers.
	etsysRadiusAcctClientServerEntry	An entry (conceptual row) representing a RADIUS Accounting server with which the client shares a secret. If RADIUS Accounting is not enabled, this table is ignored. All created conceptual rows are non-volatile and as such must be maintained upon restart of the agent.
	etsysRadiusAcctClientServerIndex	A number uniquely identifying each conceptual row in the etsysRadiusAcctClientServerTable. In the event of an agent restart, the same value of etsysRadiusAcctClientServerIndex must be used to identify each conceptual row in

Table/Group	Supported Variables	Comments
		etsysRadiusAcctClientServerTable as was used prior to the restart.
	etsysRadiusAcctClientServerAddressType	The type of Internet address by which the RADIUS Accounting server is reachable.
	etsysRadiusAcctClientServerAddress	The Internet address for the RADIUS Accounting server. Note that implementations must limit themselves to a single entry in this table per reachable server. The etsysRadiusAcctClientServerAddress may not be empty due to the SIZE restriction. Also the size of a DNS name is limited to 64 characters. This parameter value is maintained across system reboots.
	etsysRadiusAcctClientServerPortNumber	The UDP port number (1-65535) the client is using to send requests to this server. The officially assigned port number for RADIUS Accounting is 1813. This parameter value is maintained across system reboots.
	etsysRadiusAcctClientServerSecret	This object is the secret shared between the RADIUS Accounting server and RADIUS client. This parameter value is maintained across system reboots. While the 'official' MAX-ACCESS for this object is read-create, all security-conscious implementations will 'lie' on a read, and return a null-string, or something else that is fairly innocuous. The ability to read back passwords and secret encryption keys is generally a Bad Thing (tm).
	etsysRadiusAcctClientServerSecretEntered	This indicates the existence of a shared secret.
	etsysRadiusAcctClientServerRetryTimeout	The number of seconds to wait for a RADIUS Accounting Server to respond to a request. This parameter value is maintained across system reboots. A value of -1 indicates that the appropriate realm value should be used - etsysRadiusAcctClientMgmtRetryTimeout or etsysRadiusAcctClientNetworkRetryTimeout.
	etsysRadiusAcctClientServerRetries	The number of times to resend an accounting packet if a RADIUS Accounting Server does

Table/Group	Supported Variables	Comments
		not respond to a request. This parameter value is maintained across system reboots.
	etsysRadiusAcctClientServerClearTime	On a read, this value indicates the number of seconds since the counters, as defined in the IETF standard RADIUS Accounting Client MIB (RFC2618), were cleared. On a write, the client counters will be cleared and the clear time will be set to zero.
	etsysRadiusAcctClientServerStatus	Lets users create and delete RADIUS Accounting server entries on systems that support this capability. Rules 1. When creating a RADIUS Accounting Client, it is up to the management station to determine a suitable etsysRadiusAcctClientServerIndex. To facilitate interoperability, agents should not put any restrictions on the etsysRadiusAcctClientServerIndex beyond the obvious ones that it be valid and unused. 2. Before a new row can become 'active', values must be supplied for the columnar objects etsysRadiusAcctClientClientServerAddress, and etsysRadiusAcctClientServerSecret. 3. The value of etsysRadiusAcctClientServerStatus must be set to 'notInService' in order to modify a writable object in the same conceptual row. 4. etsysRadiusAcctClientServer entries whose status is 'notReady' or 'notInService' will not be used for Accounting.
	etsysRadiusAcctClientServerUpdateInterval	This indicates how many seconds elapse between accounting interim updates from the client to this RADIUS server. This parameter value is maintained across system reboots. A value of zero means no Interim Updates. If the value is less than

Table/Group	Supported Variables	Comments
		etsysRadiusAcctClientServerIntervalMinimum, the etsysRadiusAcctClientServerIntervalMinimum value will be used for the update interval time. If RADIUS Accounting is not enabled, this object is ignored. Note that Accounting Interim Updates are not issued by the RADIUS Accounting Client, unless so requested by the RADIUS Server in an Access Accept packet. A value of -1 indicates that this object is not configured and this device will use the value configured in etsysRadiusAcctClientUpdateInterval for the minimum value in seconds between accounting interim updates for this RADIUS server.
	etsysRadiusAcctClientServerIntervalMinimum	This indicates the minimum value in seconds between accounting interim updates supported by the managed entity for this RADIUS server. This parameter value is maintained across system reboots. If RADIUS Accounting is not enabled, this object is ignored. A value of -1 indicates that this object is not configured and this device will use the value configured in etsysRadiusAcctClientIntervalMinimum for the minimum value in seconds between accounting interim updates for this RADIUS server.
	etsysRadiusAcctClientServerRealmType	This object allows a server to be restricted to providing authentication services to certain classes of access methods. any(1) - the server will be available to authenticate users originating from any of the access methods. mgmtAccess(2) - the server will only be available for authenticating users that have requested management access via the console, telnet, SSH, HTTP, etc. networkAccess(3) - the server will only be available for authenticating users that are attempting to gain

Table/Group	Supported Variables	Comments
		access to the network via 802.1X, Port Web Authentication, MAC Authentication, etc. Non-default values for this object should be used when there is a desire to have one set of servers used for authenticating management access requests and a different set used for authenticating network access requests. When this object has the value of any(1) then the associated server will be in each set. The precedence order defined by the relative value of the etsysRadiusAuthServerIndex will be maintained within each set of servers.
	etsysRadiusAcctClientServerClientAddressType	This object specifies how etsysRadiusAcctClientServerClientAddress is encoded. Support for all possible enumerations defined by InetAddressType is NOT REQUIRED.
	etsysRadiusAcctClientServerClientAddress	The encoded unicast IP address of a local system interface. RADIUS requests will be sent from this address.
	etsysRadiusAcctClientServerClientVirtualRouterName	The name of the local system virtual router that traffic sent to this RADIUS server should be associated with. Writing this object with a zero length string clears the virtual router name for this server.

Table/Group	Supported Variables	Comments
Additional RADIUS Acct Client Scalars	etsysRadiusAcctClientMgmtEnable	This object indicates whether or not RADIUS Accounting is enabled or disabled for management sessions. This parameter value is maintained across system reboots. The unset value (0) indicates that the value in etsysRadiusAcctClientEnable should be used.
	etsysRadiusAcctClientNetworkEnable	This object indicates whether or not RADIUS Accounting is enabled or disabled for network sessions. This parameter value is maintained across system reboots. The unset value (0) indicates that the value in etsysRadiusAcctClientEnable should be used.
	etsysRadiusAcctClientMgmtRetryTimeout	The number of seconds to wait for a RADIUS Accounting Server in the management realm to respond to a request. This parameter value is maintained across system reboots. This value may be overridden by the per server etsysRadiusAcctClientServerRetryTimeout.
	etsysRadiusAcctClientNetworkRetryTimeout	The number of seconds to wait for a RADIUS Accounting Server in the network realm to respond to a request. This parameter value is maintained across system reboots. This value may be overridden by the per server etsysRadiusAcctClientServerRetryTimeout.

ENTERASYS-RADIUS-AUTH-CLIENT-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
Branches of the Enterasys RADIUS Auth Client MIB	etsysRadiusAuthClientMIBObjects	
RADIUS Auth Client Scalars	etsysRadiusAuthClientRetryTimeout	The number of seconds to wait for a RADIUS Server to respond to a request. Maintaining the value of this object across agent reboots is REQUIRED.
	etsysRadiusAuthClientRetries	The number of times to resend an authentication packet if a RADIUS Server does not respond to a request. Maintaining the value of this object across agent reboots is REQUIRED.
	etsysRadiusAuthClientEnable	Controls and indicates the operational state of the RADIUS client functionality. Maintaining the value of this object across agent reboots is REQUIRED.
	etsysRadiusAuthClientAuthType	This indicates which method is being used for authentication. mac(1) - indicates MAC address authentication eapol(2) - indicates EAPOL authentication This list of enumeration constants is subject to change. This parameter value is maintained across system reboots.

Table/Group	Supported Variables	Comments
RADIUS Auth Client Server Table	etsysRadiusAuthServerTable	A list of RADIUS servers that this client may attempt to use.
	etsysRadiusAuthServerEntry	A RADIUS server that this client may attempt to use.
	etsysRadiusAuthServerIndex	A number uniquely identifying each conceptual row in the etsysRadiusAuthServerTable. etsysRadiusAuthServerIndices with range between 2147483642 and 2147483647 indicate the highest priority servers stored for backwards compatibility. After consideration of these special indice values this value indicates the relative priority of the servers. Relative priority of the servers is used when using the standard authentication retransmission algorithm. Maintaining the value of etsysRadiusAuthServerIndex for all active(1) entries across agent reboots is REQUIRED.
	etsysRadiusAuthClientServerAddressType	This object specifies how etsysRadiusAuthClientServerAddress is encoded. Support for all possible enumerations defined by InetAddressType is NOT REQUIRED.
	etsysRadiusAuthClientServerAddress	The encoded unicast IP address or hostname of a RADIUS server. RADIUS requests will be sent to this address. If this address is a DNS hostname, then that hostname SHOULD be resolved into an IP address each time an authentication session is initialized.
	etsysRadiusAuthClientServerPortNumber	The UDP port number (0-65535) the client will use to send RADIUS requests to this server.
	etsysRadiusAuthClientServerSecret	This object is the secret shared between the RADIUS authentication server and the RADIUS client. On a read operation this object MUST return a zero length string. Writing this object with a zero length string clears the secret.
	etsysRadiusAuthClientServerSecretEntered	true(1) - Indicates that etsysRadiusAuthClientServerSecret

Table/Group	Supported Variables	Comments
		was last set with some value other than the empty string. false(2) - Indicates that etsysRadiusAuthClientServerSecret has never been set, or was last set to the empty string."
	etsysRadiusAuthClientServerClearTime	The number of seconds elapsed since the counters were last cleared. Writing the value zero will cause the servers counters to be cleared and the clear time will be set to zero. Writing any value other than zero will have no effect.
	etsysRadiusAuthClientServerStatus	The row status of this conceptual row in the table. active - The server is available for performing RADIUS operations. Other writable leaves in this row MUST NOT be modified while the row is in the active state. notInService - The entry is fully configured but is not available for performing RADIUS operations. Conceptual rows with this status MAY be deleted at the discretion of the agent, at which time it will be treated as if destroy(6) was SET to this object. notReady - The entry exists in the agent, but is missing information necessary in order to be available for use by the managed device (i.e., one or more required columns in the conceptual row have not been instantiated); createAndGo - Not possible. createAndWait - Creates a new instance of a conceptual row, but does not make it available for use by the managed device. destroy - This will remove the conceptual row from the table and make it unavailable for RADIUS client operations. This MUST also cause any persistent data related to this row to be removed from the system. Maintaining active(1) entries across agent reboots is REQUIRED.
	etsysRadiusAuthClientServerRealmType	This object allows a server to be restricted to providing

Table/Group	Supported Variables	Comments
		authentication services to certain classes of access methods. any(1) - the server will be available to authenticate users originating from either the mgmtAccess or networkAccess realms. mgmtAccess(2) - the server will only be available for authenticating users that have requested management access via the console, telnet, SSH, HTTP, etc. networkAccess(3) - the server will only be available for authenticating users that are attempting to gain access to the network via 802.1X, Port Web Authentication, MAC Authentication, etc. nms(4) - the server will only be available for authenticating users that are attempting to gain access to the network via network virtualization or virtual machine tracking using a network management system. This realm type is only allowed for backwards compatibility and is not included when any is specified. Non-default values for this object should be used when there is a desire to have one set of servers used for authenticating management access requests and a different set used for authenticating network access requests. When this object has the value of any(1) then the associated server will be in each of the mgmtAccess and networkAccess sets. The precedence order defined by the relative value of the etsysRadiusAuthServerIndex will be maintained within each set of servers."
	etsysRadiusAuthClientServerTimeout	The number of seconds to wait for a RADIUS Server to respond to a request. A value of -1 indicates that the server timeout specified by etsysRadiusAuthClientRetryTimeout should be used for this server. Maintaining the value of this object across agent reboots is REQUIRED.
	etsysRadiusAuthClientServerRetries	The number of times to resend an authentication packet if a RADIUS

Table/Group	Supported Variables	Comments
		Server does not respond to a request. A value of -1 indicates that the server retries specified by etsysRadiusAuthClientRetries should be used for this server. Maintaining the value of this object across agent reboots is REQUIRED.
	etsysRadiusAuthClientServerStickyMaxSessions	The maximum number of sessions associated with this server when using the sticky round robin authentication retransmission algorithm. This value is not used when other retransmission algorithms are being utilized. Sessions that are successfully authenticated with this server are considered associated. Maintaining the value of this object across agent reboots is REQUIRED.
	etsysRadiusAuthClientServerStickyCurSessions	The current number of sessions associated with this server when using the sticky round robin authentication retransmission algorithm. This value is not used when other retransmission algorithms are being utilized. Sessions that are successfully authenticated with this server are considered associated.
	etsysRadiusAuthClientServerClientAddressType	This object specifies how etsysRadiusAuthClientServerClientAddress is encoded. Support for all possible enumerations defined by InetAddressType is NOT REQUIRED.
	etsysRadiusAuthClientServerClientAddress	The encoded unicast IP address of a local system interface. RADIUS requests will be sent from this address.
	etsysRadiusAuthClientServerClientVirtualRouterName	The name of the local system virtual router that traffic sent to this RADIUS server should be associated with. Writing this object with a zero length string clears the virtual router name for this server.

Table/Group	Supported Variables	Comments
Additional RADIUS Auth Client Scalars	etsysRadiusAuthClientAttrMgmtPassword	This value indicates which method is being used to send management access passwords to the RADIUS server. standard(1) - Use the User-Password attribute at defined in RFC2865 mschapv2(2) - Use the MS-CHAP2-Response attribute as defined in RFC2548 This parameter value is maintained across system reboots.
	etsysRadiusAuthClientRetransmissionAlgorithm	This indicates which method is being used for the authentication retransmission algorithm. standard(1) - this is the legacy Enterasys authentication retransmission algorithm. It is a combination of back off where the highest priority server is always used first for every authentication transaction and round robin where if the highest priority server is unable to respond within the configured timeout period the software immediately moves to the next highest priority server in the list. roundRobin(2) - Each new authentication transaction uses the next highest priority server from the server that was initially used for the transaction prior. After the lowest priority server in the list is used the next transaction will use the highest priority server. stickyRoundRobin(3) - Each new authentication transaction uses the next highest priority server as in roundRobin except when that session is either currently associated with a server already or has been associated with a server in the past. etsysRadiusAuthClientServerStickyMaxSessions and etsysRadiusAuthClientServerStickyCurSessions values are used to determine when previously associated sessions need to be associated with less used servers. This list of enumeration constants is subject to change. This

Table/Group	Supported Variables	Comments
		parameter value is maintained across system reboots.
	etsysRadiusAuthClientMgmtRetryTimeout	The number of seconds to wait for a RADIUS Server to respond to a request when processing management sessions. Maintaining the value of this object across agent reboots is REQUIRED. A value of -1 indicates that the value in etsysRadiusAuthClientRetryTimeout should be used.
	etsysRadiusAuthClientNetworkRetryTimeout	The number of seconds to wait for a RADIUS Server to respond to a request when procesing network sessions. Maintaining the value of this object across agent reboots is REQUIRED. A value of -1 indicates that the value in etsysRadiusAuthClientRetryTimeout should be used.
	etsysRadiusAuthClientNmsRetryTimeout	The number of seconds to wait for a RADIUS Server to respond to a request when procesing NMS sessions. Maintaining the value of this object across agent reboots is REQUIRED. A value of -1 indicates that the value in etsysRadiusAuthClientRetryTimeout should be used.
	etsysRadiusAuthClientMgmtEnable	Controls and indicates the operational state of the RADIUS client functionality for management sessions. Maintaining the value of this object across agent reboots is REQUIRED. The unset value (0) indicates that the value in etsysRadiusAuthClientEnable should be used.
	etsysRadiusAuthClientNetworkEnable	Controls and indicates the operational state of the RADIUS client functionality for network sessions. Maintaining the value of this object across agent reboots is REQUIRED. The unset value (0) indicates that the value in etsysRadiusAuthClientEnable should be used.

ENTERASYS-RADIUS-DYNAMIC-AUTHOR-SERVER-EXT-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
MIB Objects	etsysRadiusDynAuthorServerMIBObjects	This object indicates whether or not RADIUS Dynamic Authorization is enabled or disabled. This parameter value MUST be maintained across system reboots.
	etsysRadiusDynAuthorServerClientTable	The (conceptual) table listing the RADIUS Accounting servers.
	etsysRadiusDynAuthorServerClientEntry	An entry (conceptual row) representing a RADIUS dynamic authorization server with which the server shares a secret. If RADIUS dynamic authorization is not enabled, this table is ignored. All created conceptual rows are non-volatile and as such MUST be maintained upon restart of the agent.
	etsysRadiusDynAuthorServerClientIndex	A number uniquely identifying each conceptual row in the etsysRadiusDynAuthorServerClient Table. In the event of an agent restart, the same value of etsysRadiusDynAuthorServerClient Index MUST be used to identify each conceptual row in etsysRadiusDynAuthorServerClient Table as was used prior to the restart.
	etsysRadiusDynAuthorServerClientAddressType	The type of Internet address by which the RADIUS Dynamic Authorization Client is reachable.
	etsysRadiusDynAuthorServerClientAddress	The Internet address for the RADIUS Dynamic Authorization Client. Note that implementations MUST limit themselves to a single entry in this table per reachable server. The etsysRadiusDynAuthorServerClient Address may not be empty due to the SIZE restriction. Also the size of a DNS name is limited to 64 characters. This parameter value is maintained across system reboots.
	etsysRadiusDynAuthorSe	This object is the secret shared

Table/Group	Supported Variables	Comments
	rverClientSecret	between the RADIUS Dynamic Authorization client and RADIUS server. This parameter value is maintained across system reboots. While the 'official' MAX-ACCESS for this object is read-create, all implementations MUST return an empty string on a read.
	etsysRadiusDynAuthorServerClientSecretEntered	true(1) - Indicates that etsysRadiusDynAuthorServerClientSecret was last set with some value other than the empty string. false(2) - Indicates that etsysRadiusDynAuthorServerClientSecret has never been set, or was last set to the empty string.
	etsysRadiusDynAuthorServerClientStatus	Lets users create and delete RADIUS Dynamic Authorization client entries on systems that support this capability. Rules 1. When creating a RADIUS Dynamic Authorization Client, it is up to the management station to determine a suitable etsysRadiusDynAuthorServerClientIndex. To facilitate interoperability, agents SHOULD not put any restrictions on the etsysRadiusDynAuthorServerClientIndex beyond the obvious ones that it be valid and unused. 2. Before a new row can become 'active', values must be supplied for the columnar objects etsysRadiusDynAuthorServerClientAddress, etsysRadiusDynAuthorServerClientSecret, etsysRadiusDynAuthorServerClientAddress, and etsysRadiusDynAuthorClientServerClientVirtualRouterName. 3. The value of etsysRadiusDynAuthorServerClientStatus must be set to 'notInService' in order to modify a writable object in the same conceptual row. 4. etsysRadiusDynAuthorServerClient entries whose status is

Table/Group	Supported Variables	Comments
		'notReady' or 'notInService' will not be used for Dynamic Authorization.
	etsysRadiusDynAuthorClientServerClientAddressType	This object specifies how etsysRadiusDynAuthorServerClientAddressType is encoded. Support for all possible enumerations defined by InetAddressType is NOT REQUIRED.
	etsysRadiusDynAuthorClientServerClientAddress	The encoded unicast IP address of a local system interface. RADIUS Dynamic Authorization responses will be sent from this address.
	etsysRadiusDynAuthorClientServerClientVirtualRouterName	The name of the local system virtual router that traffic sent to this RADIUS Dynamic Authorization server should be associated with. Writing this object with a zero length string clears the virtual router name for this server.

ENTERASYS SYSLOG MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
etsysSyslogServer table group	etsysSyslogServerMaxEntries	The maximum number of entries allowed in the etsysSyslogServerTable. Read-only.
	etsysSyslogServerNumEntries	The number of entries currently in the etsysSyslogServerTable. Read-only.
	etsysSyslogServerTableNextAvailableIndex	This object indicates the numerically lowest available index within this entity, which may be used for the value of etsysSyslogServerIndex in the creation of a new entry in the etsysSyslogServerTable. An index is considered available if the index value falls within the range of 1 to 16 and is not being used to index an existing entry in the etsysSyslogServerTable contained within this entity. A value of zero indicates that all of the entries in the etsysSyslogServerTable are currently in use. This value should only be considered a guideline for management creation of etsysSyslogServerEntries, there is no requirement on management to create entries based upon this index value. Read-only.

Table/Group	Supported Variables	Comments
etsysSyslogServer Table	etsysSyslogServerTable	A table containing Syslog server information. Not-accessible
	etsysSyslogServerEntry	Defines the information to generate syslog messages to an aggregating agent or server. Entries within this table with an access level of read-create MUST be considered non-volatile and MUST be maintained across entity resets. Not-accessible
	etsysSyslogServerIndex	A unique arbitrary identifier for this syslog server. Not-accessible
	etsysSyslogServerDescription	Administratively assigned textual description of this syslog server. Not supported
	etsysSyslogServerAddressType	The type of Internet address by which the Syslog server is specified in etsysSyslogServerAddress. Only IPv4 and IPv6 are supported for etysysSyslogServerAddressType Read-create
	etsysSyslogServerAddress	The Internet address for the Syslog message server. The use of DNS domain names is discouraged, and agent support for them is optional. Deciding when, and how often, to resolve them is an issue. Not resolving them often enough means you might lose synchronization with the associated entry in the DNS server, and resolving them too often might leave you without access to the Syslog server during critical network events. Read-create
	etsysSyslogServerUdpPort	The UDP port number the client is using to send

Table/Group	Supported Variables	Comments
		requests to this server. If an entity only supports sending messages using a single UDP port to all servers then this may optionally be implemented read-only, in which case the current value of etsysSyslogServerDefaultUdpPort will be used. Read-create
	etsysSyslogServerFacility	The syslog facility (local0-local7) that will be encoded in messages sent to this server. If an entity only supports encoding a single facility in all messages to all servers then this may optionally be implemented read-only, in which case the current value of etsysSyslogServerDefaultFacility will be used. Read-create
	etsysSyslogServerSeverity	The maximum severity level of the messages that should be forwarded to the syslog server. The higher the level, the lower the severity. If an entity only supports filtering based on a single severity level for all servers then this may optionally be implemented read-only, in which case the current value of etsysSyslogServerDefaultSeverity will be used. Read-create
	etsysSyslogServerRowStatus	This object allows for the dynamic creation and deletion of entries within the etsysSyslogServerTable as well as the activation and deactivation of these entries. When this object's value is set to notInService(2) this server will not be sent any

Table/Group	Supported Variables	Comments
		messages, nor will any of its counters be incremented. The agent should not delete a row, except in the case of the loss of persistent storage. Refer to the RowStatus convention for further details on the behavior of this object. Read-create
	etsysSyslogServerVirtualRouterName	The name of the local system virtual router that traffic sent to this syslog server should be associated with. Writing this object with a zero length string sets the virtual router name to its default value. The default value for etsysSyslogServerVirtualRouterName is VR-Mgmt. Read-create

Table/Group	Supported Variables	Comments
The Syslog Server Defaults	etsysSyslogServerDefaultUdpPort	The default UDP port number that the managed entity is using to send syslog messages. This value will be used as the default value for etsysSyslogServerUdpPort when creating rows in the etsysSyslogServerTable and either: • no value is specified for etsysSyslogServerUdpPort • etsysSyslogServerUdpPort is implemented read-only. If etsysSyslogServerUdpPort is implemented read-only, and this value is changed, it should affect the UDP port that is used to send syslog messages to all servers as soon as it is practical. This parameter value is maintained across system reboots. etsysSyslogServerDefaultUdpPort value is 514. Read-only
	etsysSyslogServerDefaultFacility	The default syslog facility (local0-local7) that will be encoded in syslog messages. This value will be used as the default value for etsysSyslogServerFacility when creating rows in the etsysSyslogServerTable and either: • no value is specified for etsysSyslogServerFacility • etsysSyslogServerFacility is implemented read-only. If etsysSyslogServerFacility is implemented read-only, and this value is changed, it should affect the syslog facility that is encoded in all syslog messages as soon as it is practical.

Table/Group	Supported Variables	Comments
		This parameter value is maintained across system reboots. etsysSyslogServerDefaultFacility value is local0(16), Read-only
	etsysSyslogServerDefaultSeverity	The default syslog message severity level that will be used to filter all syslog messages. This value will be used as the default value for etsysSyslogServerSeverity when creating rows in the etsysSyslogServerTable and either: • no value is specified for etsysSyslogServerSeverity • etsysSyslogServerSeverity is implemented read-only. The higher the severity level, the less critical it is. If etsysSyslogServerSeverity is implemented read-only, and this value is changed, it should affect the syslog message severity level that will be used to filter all syslog messages as soon as it is practical. This parameter value is maintained across system reboots. etsysSyslogServerDefaultSeverity value is debug(8). Read-only

The mapping between Syslog MIB severity (for example, etsysSyslogServerSeverity) and ExtremeXOS severity is not one-to-one as ExtremeXOS does not support “emergency” and “alert” severity; and Syslog MIB does not support “debug-data”, “debug-verbose” and “debug-summary” severity.

The following table shows the mapping between the two:

etsysSyslogServerSeverity	ExtremeXOS Syslog Severity
emergency, alert, critical	critical
Error	error

etsysSyslogServerSeverity	ExtremeXOS Syslog Severity
Warning	warning
Notice	notice
Info	info
debug	debug-data,debug-verbose, debug-summary

ENTERASYS-VLAN-AUTHORIZATION-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
etsysVlanAuthorizationEnable	All objects	The enable/disable state for the VLAN authorization feature. When disabled, no modifications to the VLAN attributes related to packet switching should be enforced.
etsysVlanAuthorizationTable	All objects	Extensions to the table that contains information about every port that is associated with this transparent bridge.
etsysVlanAuthorizationEntry	All objects	A list of extensions that support the management of proprietary features for each port of a transparent bridge. This is indexed by dot1dBasePort.
etsysVlanAuthorizationStatus	All objects	The enabled/disabled status for the application of VLAN authorization on this port, if disabled, the information returned in the VLAN-Tunnel-Type from the authentication will not be applied to the port (although it should be represented in this table).

Table/Group	Supported Variables	Comments
etsysVlanAuthorizationAdminEgress	All objects	Controls the modification of the current vlan egress list (of the vlan returned in the VLAN-Tunnel-Type, and reported by etsysVlanAuthorizationVlanID) upon successful authentication in the following manner: none(1) No egress manipulation will be made. tagged(2) The authenticating port will be added to the current egress for the VLAN-ID returned. untagged(3) The authenticating port will be added to the current untagged egress for the VLAN-ID returned. dynamic(4) The authenticating port will use information returned in the authentication response to modify the current egress lists. This value is supported only if the device supports a mechanism through which the egress status may be returned through the RADIUS response. Should etsysVlanAuthorizationEnable become disabled, etsysVlanAuthorizationStatus become disabled for a port, or should etsysVlanAuthorizationVlanID become 0 or 4095, all effect on the port egress MUST be removed.

Table/Group	Supported Variables	Comments
etsysVlanAuthorizationOperEgress	All objects	Reports the current state of modification to the current vlan egress list (of the vlan returned in the VLAN-Tunnel-Type) upon successful authentication, if etsysVlanAuthorizationStatus is enabled, in the following manner: none(1) No egress manipulation will be made. tagged(2) The authenticating port will be added to the current egress for the VLAN-ID returned. untagged(3) The authenticating port will be added to the current untagged egress for the VLAN-ID returned. The purpose of this leaf is to report, specifically when etsysVlanAuthorizationAdminEgress has been set to dynamic(4), the currently enforced egress modification. If the port is unauthenticated, or no VLAN-ID has been applied, this leaf should return none(1).
etsysVlanAuthorizationVlanID	All objects	The 12 bit VLAN identifier for a given port, used to override the PVID of the given port, obtained as a result of an authentication. A value of zero indicates that there is no authenticated VLAN ID for the given port. Should a port become unauthenticated this value MUST be returned to zero. A value of 4095 indicates that a the port has been authenticated, but that the VLAN returned could not be applied to the port (possibly because of resource constraints or misconfiguration). In this instance, the original PVID should still be applied. Should the feature become disabled or the session terminate, all effect on the Port VLAN ID MUST be removed.
etsysVlanAuthorizationGroup	All objects	A collection of objects relating to VLAN Authorization.

EXTREME-AUTOPROVISION-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeAutoProvision Config		Enables or disables the Auto Provision feature on the switch.

The following traps can be generated.

Trap	Notification Objects	Comments
extremeAutoProvisionStatus		This trap reports the auto provision result (success/failed). It contains the attributes it received from the *DHCP (Dynamic Host Configuration Protocol)* server.
	extremeAutoProvisionResult	Result of the Auto provision.
	extremAutoProvisionIpAddress	The IP address received from the DHCP server for this interface.
	extremeAutoProvisionGateway	The gateway address received from the DHCP server for this interface.
	extremeAutoProvisionTFTPServer	The IP address of the TFTP server received from the DHCP server.
	extremeAutoProvisionConfigFileName	The configuration file name received from the DHCP server which the auto-provision enabled switch will download from the TFTP server and apply. It can use either the cfg or xsf extension.

EXTREME-CFM-MIB

The ExtremeCFM MIB provides information about the Connectivity Fault Management (CFM) Group. This is an extension to IEEE8021- CFM-MIB.

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeCfmGroup	ExtremeCfmGroupOperStatus	This is a textual convention which, indicates the operational status of a group associated with a MEP on a port of an association in a given domain.
extremeCfmGroupNextIndexTable		This object contains an unused value for extremeCfmGroupIndex in the extremeCfmGroupTable, or a zero to indicate that none exists.
	extremeCfmGroupNextIndexEntry	The conceptual row of extremeCfmGroupNextIndexTable. Not accessible.
	extremeCfmGroupNextIndex	Value used as the index of the Group table entries, for this Maintenance association End Point Identifier when the management entity wants to create a new row in that table. Read only.
extremeCfmGroupTable		Includes configuration objects and operations for the Group function, mainly used by the registered clients like *ERPS (Ethernet Ring Protection Switching)*, *EAPS (Extreme Automatic Protection Switching)* to know link detection failure through CFM. Each row in the table represents a Group for the defined MEP.
	extremeCfmGroupEntry	This is a complex index with four indices to access any row in extremeCfmGroupTable. The first three are Maintenance Domain, MaNet, and MEP indices. The fourth index is the specific Group on the selected MEP. Not accessible.
	extremeCfmGroupName	The name of a CFM group. Group name must be alpha-numeric.
	extremeCfmGroupStatus	Whether the group is operational or not. Ready only.

Table/Group	Supported Variables	Comments
	extremeCfmMepIfIndex	Interface index of the interface, either a bridge port, or an aggregated IEEE 802.1 link within a bridge port, to which the MEP and hence the group is attached. Upon a restart of the system, the system, if necessary, changes the value of this variable so that it indexes the entry in the interface table with the same value of ifAlias that it indexed before the system restart. If no such entry exists, then the system sets this variable to 0. Read only.
	extremeCfmGroupRemoteMEPs	Lists the Remote MEPs associated with a group. Not all Remote MEPs of an MA may be associated with a group. Ready only.
	extremeCfmGroupClients	Lists all the registered clients with a group. The clients are informed with link failure or recovery through group status notifications. Ready only.
	extremeCfmGroupRowStatus	The status of the row. All columns must have a valid value before a row can be activated.
extremeCfmGroupMepDbTable		Maintains information about other MEPs in that group.
	extremeCfmGroupMepDbEntry	Complex index including Maintenance Domain, MaNet, MEP and Group indices along with RMEP ID to identify a row in this table. Not accessible.
	extremeCfmGroupMepDbRMepId	Maintenance association End Point Identifier of a remote MEP whose information from the group MEP database is to be returned. Not accessible.
	extremeCfmGroupMepDbRowStatus	The status of the row. All columns must have a valid value before a row can be activated.
extremeCfmGroupStatusDownUpAlarm	extremeCfmGroupStatus	A notification (DownUpAlarm) is sent to the management entity with the OID of the group that has detected the status change.

EXTREME-CLEARFLOW-MIB

This MIB defines the following Extreme-specific CLEAR-Flow traps generated by Extreme Networks devices.

Trap	Comments
extremeClearflowMessage	CLEAR-Flow message trap.
The varbinds supported in this trap are:	
extremeClearflowMsgId	User-defined message ID.
extremeClearflowMsg	User-defined message.
extremeClearflowPolicyName	Policy file name.
extremeClearflowRuleName	Rule name which triggered this message.
extremeClearflowRuleValue	Calculated rule value.
extremeClearflowRuleThreshold	Rule threshold value.
extremeClearflowRuleInterval	Rule sampling and evaluation interval.
extremeClearflowVlanName	*VLAN* name on which this policy is applied.
extremeClearflowPortName	Port name on which this policy is applied.

EXTREME-CPU-MONITOR-TOTAL-UTILIZATION-MIB

The following tables, groups, and variables are supported in this MIB:

Table/Group	Supported Variables	Comments
extremeCpuMonitorTotalUtilizationTable	extremeCpuMonitorTotalUtilizationEntry	
	extremeCpuMonitorTotalUtilizationSlotId	Read-only
	extremeCpuMonitorTotalUtilizationOfSlot	Read-only

EXTREME-EAPS-MIB

Managed objects for EAPS MIB are defined proprietary on ExtremeXOS.

Table 164 lists the existing traps supported in this MIB.

Table 164: EAPS MIB Traps

Trap	Varbinds	Comments
extremeEapsStateChange	extremeEapsName, extremeEapsMode, extremeEapsPrevState, extremeEapsState, extremeEapsFailedFlag, extremeEapsPrimaryStatus extremeEapsSecondaryStatus	Send on Master/Transit Nodes
extremeEapsLastStatusChangeTime	extremeEapsLastStatusChange extremeEapsStatusTrapCount	Send on Master/Transit Nodes. General indication of a status change using 10 second timer.
extremeEapsPortStatusChange	extremeEapsName, extremeEapsPrimaryStatus extremeEapsSecondaryStatus extremeEapsLastStatusChange	Send on Master/Transit Nodes
extremeEapsConfigChange	extremeLastConfigurationChange	30 second granularity Send on Master/Transit Nodes
extremeEapsSharedPortStateChange	extremeEapsSharedPortIfIndex extremeEapsSharedPortLinkId extremeEapsSharedPortState extremeEapsSharedPortNbrStatus extremeEapsSharedPortRootBlockerStatus extremeEapsLastStatusChange	Send on Controllerr/Partner Nodes
extremeEapsRootBlockerStatusChange	extremeEapsSharedPortIfIndex extremeEapsSharedPortRootBlockerStatus extremeEapsSharedPortRootBlockerId extremeEapsLastStatusChange	Send on Controller/Partner Nodes

Table 165 lists the MIB entries.

Table 165: EAPS MIB Entries

MIB Variable	Description
extremeEapsLastStatusChange	Time gets updated for any of the following changes... extremeEapsState, extremeEapsSharedPortState extremeEapsSharedPortSegmentStatus extremeEapsFailedFlag, extremeEapsPrimaryStatus extremeEapsSecondaryStatus extremeEapsSharedPortNbrStatus extremeEapsSharedPortRootBlockerStatus extremeEapsSharedPortSegmentFailedFlag
extremeEapsStatusTrapCount	Indicates the number of status traps sent out since the switch booted. Status traps include extremeEapsStateChange, extremeEapsPortStatusChange, extremeEapsSharedPortStateChange, extremeEapsRootBlockerStatusChange

EXTREME-EDP-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeEdpTable	All objects	This table contains *EDP (Extreme Discovery Protocol)* information of this device.
extremeEdpNeighborTable	All objects	This table contains EDP neighbor information.
extremeEdpPortTable	All objects	

EXTREME-ENTITY-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variable	Comments
extremeEntityFRUTable	entPhysicalIndex	A table containing information about each FRU based on the Entity MIB.
	extremeEntityFRUStartTime	First recorded start time.

Table/Group	Supported Variable	Comments
	extremeEntityFRUOdometer	Number of time units in service.
	extremeEntityFRUOdometerUnit	Time unit used to represent value reported by extremeEntityFRUOdometer. Depends on the underlying hardware capability.

EXTREME-ERPS-MIB

Managed objects for *ERPS* MIB are defined proprietary on ExtremeXOS. Groups and tables are implemented as read only.

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeErpsProtectedVlanTable	All objects	Contains the grouping of set of protected VLANs.
extremeErpsRingTable	All objects	Contains information for each ERPS ring present in the switch.
extremeErpsStatsTable	All objects	Contains statistics information for each of the ring present in the switch.
extremeErpsGlobalInfo		Contains the information of ERPS configured globally in the switch.
extremeErpsNotification		Contains two types of traps: extremeErpsStateChangeTrap and extremeErpsFailureTrap. extremeErpsStateChangeTrap is generated on the following events: • Local SF is received for the ring. • Local Clear SF is received for the ring. • Remote failure is detected on this ring. • Remote failure is cleared on this ring. • Force Switch is issued for a ring. • Manual Switch is issued for a ring.

EXTREME-ESRP-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeEsrpDomainTable	All objects	This table contains information for domains on this device.
extremeEsrpDomainMemberTable	All objects	This table contains information for member VLANs of the ESRP domain.
extremeEsrpDomainNeighborTable	All objects	This table contains neighbor router information for ESRP domains on this device.
extremeEsrpDomainAwareTable	All objects	This table contains ESRP aware information for this device.
extremeEsrpDomainStatsTable	All objects	This table contains statistics on ESRP hello packets exchanged and ESRP state changes for this device.
extremeEsrpNotifications	extremeEsrpDomainStateChange	Signifies ESRP state change.

EXTREME-FDB-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeFdbIpFdbTable	All objects	Supports *SNMP* get and get next operations only.
extremeFdbPermFdbTable	All objects	

Table/Group	Supported Variables	Comments
extremeFdbMacExosFdbTable	extremeFdbMacExosFdbEntry	A table that contains information about the hardware MAC *FDB (forwarding database)* table. Supported only on switches running ExtremeXOS. Supports SNMP get and get next operations only.
extremeFdbMacFdbCounterTable	All objects	Supports SNMP get and get next operations only.

Trap	Comments
extremeMACTrackingAdd	The specified MAC address was added to the FDB on the mentioned port and VLAN.
extremeMACTrackingDel	The specified MAC address was deleted from the FDB on the mentioned port and VLAN.
extremeMACTrackingMove	The specified MAC address was moved from the previous port to the new port on the specified VLAN.

EXTREME-MAC-AUTH-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeMacAuthClientTable	All objects	This table contains client authentication based on the source MAC address of the traffic received on a port.
ExtremeMacAuthClientEntry		An entry in the MAC authentication table. Each entry represents a MAC authentication client
	extremeMacAuthClientAddress	The MAC address of the client.
	extremeMacAuthClientInitialize	Remove the entry for the client from extremeMacAuthClientTable.
	extremeMacAuthClientReauthenticate	Re-authenticate the client on all ports on which the client is connected.

EXTREME MLAG MIB

The following tables, groups, and variables are supported in this MIB.

Note
EXTREME MLAG MIB objects such as extremeMlagLocalAddrType, extremeMlagLocalIP, extremeMlagAlternateLocalAddrType, extremeMlagAlternateLocalIP are defined with a maximum access of read-create and are populated automatically based on MLAG peer info.

Table/Group	Supported Variables	Comments
extremeMlagPeerTable		This table contains MLAG peer entries. This table contains a MLAG peer switch association structure, associates an MLAG peer structure with an MLAG peer switch IP address. Displays User configured items, MLAG peer switch state, MLAG port count and health-check statistics.
	extremeMlagPeerName	Read-create The name of the MLAG Peer switch. MLAG Peer names must be alpha-numeric and must start with an alpha character.
	extremeMlagPeerVlan	Read-only The name of the VLAN through which the MLAG Peer can be reached.
	extremeMlagPeerVR	Read-create The name of the VR with which the MLAG peer VLAN is associated with.
	extremeMlagLocalAddrType	Read-create The IP address type of the extremeMlagLocalIP object.
	extremeMlagLocalIP	Read-create Local Switch IPAddress of ISC connection.
	extremeMlagPeerAddrType	Read-create The IP address type of the extremeMlagPeerIP object.
	extremeMlagPeerIP	Read-create MLAG peer IP address for the ISC. The specified IP address must be contained within an existing direct route.

Table/Group	Supported Variables	Comments
	extremeMlagPeerPortCount	Read-only Number of MLAG Ports associated with this MLAG peer
	extremeMlagPeerCheckPointStatus	Read-only Checkpointing Status of this MLAG peer
	extremeMlagPeerRxHellos	Read-only Number of Hellos received from the peer
	extremeMlagPeerRxCheckpointMsgs	Read-only Number of Checkpoint Messages received from the peer
	extremeMlagPeerHelloErrors	Read-only Number of Hello Error Messages
	extremeMlagPeerHelloTimeouts	Read-only Number of Hello Timeout Messages
	extremeMlagPeerUptime	Read-only Specifies the time that the Connectivity with the peer is up
	extremeMlagPeerLocalTxInterval	Read-create Length of the time in milli seconds between health check hello packets transmitted by this switch.
	extremeMlagPeerRemoteTxInterval	Read-only Transmitting Hello Interval of Peer Switch in milli seconds
	extremeMlagPeerTxHellos	Read-only Number of Health Check Hellos transmitted
	extremeMlagPeerTxCheckpoints	Read-only Number of transmitted Checkpoint messages
	extremeMlagPeerCheckpointErrors	Read-only Number of Checkpoint Errors
	extremeMlagPeerConnnectErrors	Read-only Number of Peer Connect errors
	extremeMlagPeerRowStatus	Read-create The status of this entry

Table/Group	Supported Variables	Comments
	extremeMlagPeerCfgLacpMac	Read-create This MAC address specifies the MLAG LACP MAC configured on this switch. If configured, this MAC is used as the System Identifier in the LACPDUs sent over the MLAG Ports. If not configured, the MAC address of MLAG peer having highest IP address for ISC control VLAN is chosen as the System Identifier in the LACPDUs sent over the MLAG ports.
	extremeMlagPeerOperLacpMac	Read-only This MAC address specifies the MLAG LACP MAC that is being used as the System Identifier in the LACPDUs sent over the MLAG Ports. This may be different from extremeMlagPeerCfgLacpMac if same extremeMlagPeerCfgLacpMac is not configured on both the MLAG peers.
	extremeMlagPeerAlternateVlan	Read-only The name of the alternate VLAN through which the MLAG Peer can be reached when the primary path is unreachable.
	extremeMlagPeerAlternateVR	Read-create The name of the alternate VR with which the MLAG peer alternate VLAN is associated with.
	extremeMlagAlternateLocalAddrType	Read-create The IP address type of the extremeMlagAlternateLocalIP object.
	extremeMlagAlternateLocalIP	Read-create Local Switch IPAddress for alternate health check connection to MLAG peer.
	extremeMlagAlternatePeerAddrType	Read-create The IP address type of the extremeMlagAlternatePeerIP object.
	extremeMlagAlternatePeerIP	Read-create MLAG peer IP address for alternate health check connection.

Table/Group	Supported Variables	Comments
	extremeMlagPeerAlternateRxHellos	Read-only Number of Hellos received from the peer on the alternate health check connection
	extremeMlagPeerAlternateHelloErrors	Read-only Number of Hello Error Messages on the alternate health check connection
	extremeMlagPeerAlternateHelloTimeouts	Read-only Number of Hello Timeout Messages on the alternate health check connection
	extremeMlagPeerAlternateTxHellos	Read-only Number of Health Check Hellos transmitted on the alternate health check connection
	extremeMlagPeerCheckPointAuthType	Read-create Enumerates the possible types of authentication types for the MLAG checkpoint connection to the MLAG peer. none(1) Do not use authentication. md5(2) Use MD5 authentication.
	extremeMlagPeerCheckpointAuthKey	Read-create The authentication key to be used for checkpoint conntection to the peer. Recommended to be of size 12-24 per RFC 3562. This object will return a blank string of minimum length when queried.
extremeMlagPortTable		This table contains Mlag port entries Binds a local port or LAG port to an MLAG port uniquely indentified by the MLAG Id value. Once the MLAG group binding is made, any change to load sharing on MLAG ports is disallowed. Displays User configured items, MLAG peer switch state, port count and health-check statistics. **Note:** To create a row in the extremeMlagPortTable, extremeMlagPortId and extremeMlagPortPeer should be specified in the SNMP SET request in addition to extremeMlagPortLocalPortIfIndex.

Table/Group	Supported Variables	Comments
	extremeMlagPortLocalPortIfIndex	Read-create The ifIndex of the local member port of the MLAG port. The specified port ifIndex may be a single port or the master port of a load share port but should not be a load share member port.
	extremeMlagPortId	Read-create The identifier which is used to identify the MLAG port
	extremeMlagPortPeer	Read-create This object identifies the MLAG peer switch with which this port is associated
	extremeMlagPortLocalLinkStatus	Read-only Local MLAG Port Status, that can be Active/Disabled/Ready/Port not present. It reflects the status of entire LAG when LAG is used in conjunction with MLAG.
	extremeMlagPortRemoteLinkStatus	Read-only Remote MLAG Port Status, which can be Up(1) - One or more links are active on the remote switch or Down(2)- No links are active on the remote switch or N/A(3) - The peer has not communicated link state for this MLAG group. It reflects the status of entire LAG when LAG is used in conjunction with MLAG.
	extremeMlagPortPeerState	Read-only MLAG Peer Switch State, up/down
	extremeMlagPortLocalFailures	Read-only Gives the number of ports that are down in the local MLAG port.
	extremeMlagPortRemoteFailures	Read-only Gives number of ports that are down in the remote MLAG port
	extremeMlagPortRowStatus	Read-create The status of this entry.

Table/Group	Supported Variables	Comments
	extremeMlagConvergenceControl	Read-write This object specifies if an ACL can be used on an MLAG port to achieve faster convergence. If an ACL is used, better convergence times can be achieved irrespective of the number of FDB entries present on the MLAG port. The second time can be chosen when convergence time is not an issue and ACLs are to be preserved.
	extremeMlagReloadDelayInterval	Read-write Time Interval in seconds that is used to delay the enabling of MLAG ports during startup. This is used to allow convergence of protocols and for reachability of MLAG peer.
	extremeMlagReloadDelayEnable	Read-write Setting this object to true will enable the reload delay feature. Enabling reload delay will delay enabling of MLAG ports during startup to allow convergence of protocols and for reachability of MLAG peer.
	extremeMlagLinkupIsolationEnable	Read-write Setting this object to true will isolate MLAG ports from sending traffic to local ISC port during link-up transition until remote ISC port is configured.
	extremeMlagPeerUp	This trap is sent when the MLAG peer comes up.
	extremeMlagPeerDown	This notification is sent when health check messages are not received from the peer MLAG switch for 3 times the configured hello interval.
	extremeMlagAltPathUp	This notification is sent when the ISC goes down and the alternate path to MLAG peer comes up.
	extremeMlagAltPathDown	This notification is sent when the alternate path to MLAG peer is down. This can happen when either alternate path or remote peer is down or the ISC link has come up.

EXTREME-MPLS-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeMplsNotifications	extremepwUpDownNotification Enable	If this object is set to true (1), then it enables the emission of pwUp and pwDown notifications; otherwise these notifications are not emitted.
	extremepwDeletedNotificationEnable	If this object is set to true (1), then it enables the emission of pwDeleted notification; otherwise this notification is not emitted.
	pwNotificationMaxRate	This variable indicates the maximum number of notifications issued per second. If events occur more rapidly, the implementation may simply fail to emit these notifications during that period, or might queue them until an appropriate time. A value of 0 means no throttling is applied and events might be notified at the rate at which they occur.
	extremepwNotificationPwIndex	This variable indicates the index of the PW that either went up, down, or was deleted as reported by the corresponding notification.
	extremepwNotificationPwOperStatus	This variable is used to report the value of pwOperStatus in pwTable (RFC 5601) associated with the PW that went up or down.
	extremepwNotificationPeerAddrType	Denotes the address type of the peer node.
	extremepwNotificationPeerAddr	This object contains the value of the peer node address.
	extremeMplsNotifTunnelIndex	Uniquely identifies a set of tunnel instances between a pair of ingress and egress LSRs. Contains part of the index of a TE tunnel that underwent state change. The same tunnel can also be looked up in the TE MIB.
	extremeMplsNotifTunnelInstance	Uniquely identifies a particular instance of a tunnel between a pair of ingress and egress LSRs.
	extremeMplsNotifTunnelIngress LSRId	Identity of the ingress LSR associated with this tunnel instance.

Table/Group	Supported Variables	Comments
	extremeMplsNotifTunnelEgressLSRId	Identity of the egress LSR associated with this tunnel instance.
	extremeMplsNotifTunnelAdminStatus	Reports the desired operational status of this tunnel.
	extremeMplsNotifTunnelOperStatus	Reports the actual operational status of this tunnel, which is typically but not limited to, a function of the state of individual segments of this tunnel.
	extremeMplsNotifLdpEntityLdpId	The LDP identifier. This is the primary index to identify a row in the mplsLdpEntityTable (RFC 3815).
	extremeMplsNotifLdpEntityIndex	This index is used as a secondary index to uniquely identify a row in the mplsLdpEntityTable (RFC 3815). This object is meaningful to some, but not all, LDP implementations.
	extremeMplsNotifLdpPeerLdpId	The LDP identifier of this LDP Peer.
	extremeMplsNotifLdpSessionState	The current state of the session, all of the states 1 to 5 are based on the state machine for session negotiation behavior.
	extremeMplsNotifLdpSession DiscontinuityTime	The initial value of this object is the value of sysUpTime when the entry was created in this table. Subsequent notifications report the time when the session between a given entity and peer goes away or a new session is established.
	extremeVplsNotifConfigIndex	Unique index for the conceptual row identifying a VPLS service in the vplsConfigTable.
	extremeVplsNotifConfigVpnId	This objects indicates the IEEE 802-1990 VPN ID of the associated VPLS service. This object has the same value as vplsConfigVpnId in the vplsConfigTable for an index value equal to extremeVplsNotifConfigIndex sent in the notification.

Table/Group	Supported Variables	Comments
	extremeVplsNotifConfigAdminStatus	The administrative state of the VPLS service. This object has the same value as vplsConfigAdminStatus in the vplsConfigTable for an index value equal to extremeVplsNotifConfigIndex sent in the notification.
	extremeVplsNotifStatusOperStatus	The current operational state of this VPLS service. A value of up (1) indicates that all PWs for this VPLS are up and the attachment circuit is up. A value of degraded (2) indicates that at least one PW for this VPLS is up and the attachment circuit is up. A value of down (3) indicates that all PWs for this VPLS are down or the attachment circuit is down.
	extremePwStatusChange	This notification is generated when the pwOperStatus object for a pseudowire transitions from up (1) to down (2) or from down (2) to up (1).
	extremePwDeleted	This notification is generated when a PW has been deleted.
	extremeMplsTunnelStatusChange	This notification is generated when the mplsTunnelOperStatus object for a TE-LSP transitions from up (1) to down (2) or from down (2) to up (1). This new state is indicated by the included value of mplsTunnelOperStatus.
	extremeMplsLdpSessionStatus Change	This notification is generated when the value of 'mplsLdpSessionState' (RFC3815) enters or leaves the operational (5) state.

Table/Group	Supported Variables	Comments
	extremeVplsStatusChange	This notification is generated to inform recipients of the state of the VPLS. When all PWs in this VPLS are up or ready and the attachment circuit is up, extremeVplsNotifStatusOperStatus is set to vplsOperStatusUp (1) in the notification. When at least one PW in this VPLS is up or ready and the attachment circuit is up, extremeVplsNotifStatusOperStatus is set to vplsOperStatusDegraded (2) in the notification. When all PWs in this VPLS are down or the attachment circuit is down, extremeVplsNotifStatusOperStatus is set to vplsOperStatusDown (3) in the notification. Once a notification has been sent with vplsOperStatusDegraded (2), no further notification will be sent until extremeVplsNotifStatusOperStatus transitions to vplsOperStatusUp (1) or vplsOperStatusDown (3).
extremeVplsConfigTable		This table specifies information for configuring and monitoring VPLS.
	extremeVplsConfigEntry	A row in this table represents a VPLS in a packet network. It is indexed by extremeVplsConfigIndex, which uniquely identifies a single VPLS.
ExtremeVplsConfigEntry	extremeVplsConfigIndex	Unique index for the conceptual row identifying a VPLS service.
	extremeVplsConfigRedunType	This variable indicates the redundancy type for this VPLS. Redundancy states can be None, *EAPS*, *ESRP (Extreme Standby Router Protocol)* and *STP (Spanning Tree Protocol)*.

Table/Group	Supported Variables	Comments
	extremeVplsConfigEAPSStatus	This variable indicates the EAPS redundancy status for this VPLS. It displays Protected if the *VLAN* is protected, else Not Protected. EAPS status displays the following values: Waiting—if waiting on EAPS status.Connected—if EAPS ring status is connected to VLAN. PW is not installed in this state.Disconnected—if EAPS ring state is disconnected. PW is installed in this state.Unknown—if EAPS is not found.
	extremeVplsConfigESRPState	This variable indicates the ESRP redundancy state for this VPLS. ESRP states can have two values: master and slave. PW is installed only when ESRP is in the master state.
extremeVplsStatusTable		This table specifies information for configuring and monitoring VPLS.
	extremeVplsStatusEntry	A row in this table represents a VPLS in a packet network. It is indexed by extremeVplsConfigIndex, which uniquely identifies a single VPLS.
	extremeVplsStatusIndex	Unique index for the conceptual row identifying a VPLS service.
	extremeVplsOperStatus	This variable indicates the VPLS operation status. This is different from the VPLS PW status shown in the standard MIB. Supported values are: Up—all PWs are in the Up or Ready state.Degraded—At least one PW is in the Up or Ready state.Down—None of the PWs are in Up or Ready state.
extremeVplsPeerTable		This table provides information for monitoring VPLS peer entries.
	extremeVplsPeerEntry	This entry contains information for all the peers belonging to a particular VPLS.
	extremeVplsIndex	Unique index for the conceptual row identifying a VPLS service.

Table/Group	Supported Variables	Comments
	extremeVplsPwBindIndex	Secondary Index for the conceptual row identifying a PW within the PwEntry which must match an entry from the PW-STD-MIB's PwTable, which represents an already-provisioned PW that is then associated with this VPLS instance.
	extremePwInstalled	Boolean true or false, indicating if PW is installed.
	extremePwMode	The peer mode of this PW. It indicates its current mode and the mode of the switch it is connected to. It can have the following values: Core-to-core, Core-to-Spoke, and Spoke-to-Core.
	extremePwConfiguredRole	The configured role of this PW. It applies only in the case the PWMode is Core-to-core. In this case, the configured role can either be Primary or Secondary. For all the other PW modes, the configured role does not apply.

The following traps can be generated:

Trap	Comments
pwStatusChange	This notification is generated when the pwOperStatus object for a PW transitions from up (1) to down (2) or from down (2) to up (1).
pwDeleted	This notification is generated when a PW has been deleted.
extremeMPLSTrapsEnable	MPLS trap status. If enabled then all the MPLS related traps are sent out.
extremeL2VpnTrapsEnable	L2VPN trap status. If enabled then all the L2VPN related traps are sent out.

EXTREME-MPLS-TE-MIB

The following tables, groups, and variables are supported in this MIB:

Table/Group	Supported Variables	Comments
extremeMplsTunnelTable	mplsTunnelIndex	The mplsTunnelTable (see RFC3812) allows new MPLS tunnels to be created between an LSR and a remote endpoint, and existing tunnels to be reconfigured or removed. Note that only point-to-point tunnel segments are supported, although multipoint-to-point and point- to-multipoint connections are supported by an LSR acting as a cross-connect. Each MPLS tunnel can thus have one out-segment originating at this LSR and/or one in-segment terminating at this LSR. Extreme Networks MPLS implementation allows tunnel instances with a common endpoint to be grouped at the ingress LSR to provide redundancy. The role of each tunnel in the group must be configured and is indicated by extremeMplsTunnelRedundancyType.
	mplsTunnelInstance	Uniquely identifies a particular instance of a tunnel between a pair of ingress and egress LSRs. It is useful to identify multiple instances of tunnels for the purposes of backup and parallel tunnels.
	mplsTunnelIngressLSRId	Identity of the ingress LSR associated with this tunnel instance.
	mplsTunnelEgressLSRId	Identity of the egress LSR associated with this tunnel instance.
	mplsTunnelRedundancyType	Identifies the tunnel redundancy type associated with this tunnel instance. A value of primary (1) or secondary (2) MAY be assigned by the network administrator or by an *SNMP* manager at the time of setting up the tunnel.

Table/Group	Supported Variables	Comments
	mplsTunnelRedundancyStatus	Indicates the actual redundancy status of this tunnel. When the status is active, the tunnel is the preferred tunnel in the group.
	mplsTunnelTransportStatus	Indicates the type of traffic the tunnel group can be used for sending. When the status is allowAllIp (0), IP traffic destined for all IPv4 routes is allowed over any tunnel in the group marked active.When the status is allowAssignedIpOnly (1), IP traffic destined only for IPv4 static routes that have been explicitly configured to use this tunnel group is allowed.When the status is allowAllLayer 2Vpn (2), Layer 2 VPN traffic for all Layer 2 VPNs is allowed over any tunnel in the group marked active.When the status is allowAssignedLayer 2VpnOnly (3), Layer 2 VPN traffic destined only for PWs that have been explicitly configured to use this tunnel group are allowed.

EXTREME-OSPF-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeOspfInterfaceTable	All objects	This table contains Extreme Networks specific information about *OSPF (Open Shortest Path First)* interfaces.

EXTREME-PoE-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremePethSystemAdminEnable		Objects are supported read-only.
extremePethSystemDisconnectPrecedence		
extremePethSystemUsageThreshold		
extremePethSystemPowerSupplyMode		

Table/Group	Supported Variables	Comments
extremePethPseSlotTable	extremePethOperatorLimit	
	extremePethPortResetPower	
extremePethPseSlotTable	All other objects	Objects are supported read-only
extremePethPortDetectionType		
extremePethPortMeasuredVoltage		
extremePethPortMeasuredCurrent		
extremePethPortFastPoeEnable		
extremePethSlotPerpetualPoe		

EXTREME-PORT-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremePortLoadshareTable	All objects	Not supported.
extremePortSummitlinkTable	All objects	Not supported.

Table/Group	Supported Variables	Comments
extremePortLoadshare2Table	extremePortLoadshare2Entry	A table of bindings between a master port and its load-sharing slaves. Create/delete entries here to add/delete a port to/from a load-sharing group. Default is empty table. There are restrictions on what row creates will be accepted by each device.
	extremePortLoadshare2MasterIf Index	The if Index value which identifies the port controlling a load-sharing group of ports that includes extremePortLoadshareSlaveIfIndex.
	extremePortLoadshare2SlaveIfIndex	The if Index value which identifies the port which is a member of a load-sharing group controlled by extremePortLoadshare2MasterIfInde.
	extremePortLoadshare2Algorithm	This value identifies the load sharing algorithm to be used for this group of load shared ports.
	extremePortLoadshare2Status	The row status variable, used according to row installation and removal conventions.
	extremePortLoadshare2MinLinks	The minimum active links that must be up in order trunk to remain up.
extremePortRateShapeTable	All objects	Not supported

Table/Group	Supported Variables	Comments
extremePortUtilizationTable	extremePortUtilizationEntry	Global *QoS (Quality of Service)* profiles are defined in the extremeQosProfileTable. This table contains a list of ports for which certain QoS parms are reported.
	extremePortUtilizationAvgTxBw	The reported average bandwidth in the transmit direction. When displayed, it shows as an integer value. For example, 99.99% is displayed as 9999.
	extremePortUtilizationAvgRxBw	The reported average bandwidth in the receive direction. When displayed, it shows as an integer value. For example, 99.99% is displayed as 9999.
	extremePortUtilizationPeakTxBw	The reported peak bandwidth in the transmit direction. When displayed it shows as an integer value. For example, 99.99% is displayed as 9999.
	extremePortUtilizationPeakRxBw	The reported peak bandwidth in the receive direction. When displayed it shows as an integer value. For example, 99.99% is displayed as 9999.
extremePortInfoTable	All objects	Not supported.
extremePortXenpakVendorTable	All objects	Not supported.
extremePortIngressStatsPortTable	All objects	Not supported.
extremePortIngressStatsQueue Table	All objects	Not supported.
extremePortEgressRateLimitTable	All objects	Not supported.
extremeWiredClientTable	All objects	Not supported.

Table/Group	Supported Variables	Comments
extremePortUtilizationExtnTable	extremePortUtilizationExtnEntry	Global QoS profiles are defined in the extremeQosProfileTable. This table contains a list of ports for which certain QoS parameters are reported.
	extremePortUtilizationAvgTxPkts	The reported number of average packets in the transmit direction per second.
	extremePortUtilizationAvgRxPkts	The reported number of average packets in the receive direction per second.
	extremePortUtilizationPeakTxPkts	The reported number of peak packets in the transmit direction per second.
	extremePortUtilizationPeakRxPkts	The reported number of peak packets in the receive direction per second.
	extremePortUtilizationAvgTxBytes	The reported number of average bytes in the transmit direction per second.
	extremePortUtilizationAvgRxBytes	The reported number of average bytes in the receive direction per second.
	extremePortUtilizationPeakTxBytes	The reported number of peak bytes in the transmit direction per second.
	extremePortUtilizationPeakRxBytes	The reported number of peak bytes in the receive direction per second.

Table/Group	Supported Variables	Comments
extremePortQosStatsTable	extremePortQosStatsEntry	This table lists Ports QoS information for either ingress or egress.
	extremePortQosIngress	Indicates whether the port is in ingress/egress.
	extremePortQP0TxBytes	The number of QoS 0 bytes that gets transmitted from this port.
	extremePortQP0TxPkts	The number of QoS 0 packets that gets transmitted from this port.
	extremePortQP1TxBytes	The number of QoS 1 bytes that gets transmitted from this port.
	extremePortQP1TxPkts	The number of QoS 1 packets that gets transmitted from this port.
	extremePortQP2TxBytes	The number of QoS 2 bytes that gets transmitted from this port.
	extremePortQP2TxPkts	The number of QoS 2 packets that gets transmitted from this port.
	extremePortQP3TxBytes	The number of QoS 3 bytes that gets transmitted from this port.
	extremePortQP3TxPkts	The number of QoS 3 packets that gets transmitted from this port.
	extremePortQP4TxBytes	The number of QoS 4 bytes that gets transmitted from this port.
	extremePortQP4TxPkts	The number of QoS 4 packets that gets transmitted from this port.
	extremePortQP5TxBytes	The number of QoS 5 bytes that gets transmitted from this port.
	extremePortQP5TxPkts	The number of QoS 5 packets that gets transmitted from this port.
	extremePortQP6TxBytes	The number of QoS 6 bytes that gets transmitted from this port.
	extremePortQP6TxPkts	The number of QoS 6 packets that gets transmitted from this port.
	extremePortQP7TxBytes	The number of QoS 7 bytes that gets transmitted from this port.
	extremePortQP7TxPkts	The number of QoS 7 packets that gets transmitted from this port.

Table/Group	Supported Variables	Comments
extremePortMauTable	All objects	Port Optics Status Table. extremePortMauTable is not supported. Use the ifMauTable (MAU-MIB) to get optics information.
extremePortCongestionStatsTable	extremePortCongestionStatsEntry	This table lists ports congestion information.
	extremePortCongDropPkts	The number of packets dropped due to congestion on this port.
extremePortQosCongestionStatsTable	extremePortQosCongestionStats Entry	This table lists ports per QoS congestion information.
	extremePortQP0CongPkts	The number of QoS 0 packets that got dropped due to congestion on this port.
	extremePortQP1CongPkts	The number of QoS 1 packets that got dropped due to congestion on this port.
	extremePortQP2CongPkts	The number of QoS 2 packets that got dropped due to congestion on this port.
	extremePortQP3CongPkts	The number of QoS 3 packets that got dropped due to congestion on this port.
	extremePortQP4CongPkts	The number of QoS 4 packets that got dropped due to congestion on this port.
	extremePortQP5CongPkts	The number of QoS 5 packets that got dropped due to congestion on this port.
	extremePortQP6CongPkts	The number of QoS 6 packets that got dropped due to congestion on this port.
	extremePortQP7CongPkts	The number of QoS 7 packets that got dropped due to congestion on this port.

Table/Group	Supported Variables	Comments
extremePortConfigTable		Not-accessible. This table lists port entries with information including auto negotiation, configured port speed, and duplex.
	extremePortConfigEntry	Not-accessible. An individual Port Information Entry with information including auto negotiation, configured port speed, and duplex.
	extremePortAutoNegotiation	Read-write access. Indicates whether the port is operating auto negotiation mode.
	extremePortAdminSpeed	Read-write access. The configured speed of the port. The current actual speed of the port can be determined from ifSpeed
	extremePortDuplex	Read-write access. Indicates whether the port is operating in half-duplex, full-duplex, or auto negotiation mode.
	extremePortMedium	Read-write access. Medium-specific attributes for combination ports.

The following traps can be generated:

Trap	Comments
extremePortMauChangeTrap	This trap is sent whenever a MAU is inserted or removed. When the MAU is inserted, the value of extremePortMauStatus is 'inserted' and extremePortMauType indicates the type of the MAU inserted. If MAU is removed, the value of extremePortMauStatus is empty and the type of the MAU will be NONE.
extremeRateLimitExceededAlarm	This notification indicates the first time a poll of a rate-limited port has a non-zero counter.

Pseudowire LSP Sharing MIB

By implementing the tables in PW LSP sharing MIB, the *SNMP* manager is able to observe the transmit packet counters over each LSP that is configured for used by the

PW, and aggregated transmit and receive packet counters over PW itself. It consist three tables:

- extremePwPerfTable
- extremePwLspOutboundMappingTable
- extremePwLspPerfTable

  There are no standard MIB tables and scalars objects that can be used for Pseudo Wire LSP Sharing. MIB table and object defined here are Extreme Networks proprietary. These three table are implemented in code/ l3protocol/ mpls/src/extrememplsmib.my.

  A new SNMP MIB object is introduced for PW LSP Load Sharing: extremePwObjects OBJECT IDENTIFIER ::= { extremeMplsMIB 5 }

extremePwPerfTable

This table contains the aggregated transmit and receive packet counters for in-service PWs.

MIB Object	Description	Variable Type	Support Status
pwPerfInPackets	Receive packet counter per PW	Read	Supported
pwPerfInBytes	Receive packet counter in bytes per PW	Read	Supported
pwPerfOutPackets	Transmit packet counters per PW	Read	Supported
pwPerfOutBytes	Transmit packet counter in bytes per PW	Read	Supported
pwPerfInHCPackets	Receive high capacitypacket counter per PW	Read	Supported
pwPerfInHCBytes	Receive packet counter in bytes per PW	Read	Supported
pwPerfOutHCPackets	Transmit packet counters per PW	Read	Supported
pwPerfOutHCBytes	Transmit packet counter in bytes per PW	Read	Supported

extremePwLspOutboundMappingTable

This table provides the mapping between PWs and LSPs by providing an LSP index. LSP indexes are assigned uniquely for each PW. Entries in this table indicate that an

LSP is being used by an in-service PW. An SNMP notification will be sent when an entry is added or deleted.

MIB Object	Description	Variable Type	Support Status
lspIndex	LSP index per LSP used by the PW. It is only unique per PW peer.	Index	Supported
pwLspOutboundLsrXcIndex	Corresponding entry in the *MPLS (Multiprotocol Label Switching)*-LSR-STD-MIB	Read	Supported
pwLspOutboundTunnelIndex	Corresponding RSVP-TE tunnel index based on LSP index used by the PW	Read	Supported
pwLspOutboundTunnelInstance	Corresponding RSVP-TE tunnel instance based on LSP index used by the PW	Read	Supported
pwLspOutboundTunnelLclLSR	Corresponding RSVP-TE tunnel ingress LSR ID based on LSP index used by the PW	Read	Supported
pwLspOutboundTunnelPeerLSR	Corresponding RSVP-TE tunnel egress LSR ID based on LSP index used by the PW	Read	Supported
pwLspOutboundTunnelTypeInUse	LSP Type used by the PW. mplsTe value (MIB definition is 1) is being used for RSVP-TE LSPs, and mplsNonTe value (MIB definition is 2) is being used for Static or LDP LSPs.	Read	Supported

extremePwLspPerfTable

This table contains the transmit packet and byte counters for traffic sent over a specific PW using a specific LSP.

MIB Object	Description	Variable Type	Support Status
pwLspPerfOutPackets	Receive packet counter per LSP used by the PW.	Index	Supported
pwLspPerfOutBytes	Receive packet counter in bytes per LSP used by the PW.	Read	Supported
pwLspPerfOutHCPackets	Receive high capacity packet counter per LSP used by the PW.	Read	Supported
pwLspPerfOutHCBytes	Receive high capacity packet counter in bytes per LSP used by the PW.	Read	Supported

EXTREME-QOS-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeQosProfileTable	extremeQosProfileIndex	An index that uniquely identifies an entry in the *QoS* table.
	extremeQosProfileName	A unique QoS profile name.
	extremeQosProfileMinBw	The minimum percentage of bandwidth that this queue requires. The switch is required to provide the minimum amount of bandwidth to the queue. The lowest possible value is 0%.
	extremeQosProfileMaxBw	The maximum percentage of bandwidth that this queue is permitted to use.
	extremeQosProfilePriority	The level of priority at which this queue will be serviced by the switch.

Table/Group	Supported Variables	Comments
	extremeQosProfileRowStatus	The status of the extremeQosProfile entry. This object can be set to: active (1)createAndGo (4)createAndWait(5)destroy (6) The following values may be read: active (1)notInService(2)notReady (3)
extremeIQosProfileTable	extremeIQosProfileIndex	The ExtremeXOS software does not support global QoS profile settings in CLI; it supports per port settings only. Walks of this table display the default values with which the ports are initialized.
	extremeIQosProfileName	A unique ingress QoS profile name.
	extremeIQosProfileMinBwType	The type of the current minimum bandwidth setting. A value of 1 signifies that the minimum bandwidth value is a percentage of the configurable port bandwidth. A value of 2 or 3 signifies a guaranteed minimum available bandwidth in Kbps or Mbps, respectively.
	extremeIQosProfileMinBw	The guaranteed minimum bandwidth for this queue, expressed as either a percentage or a specific bandwidth value, as specified by the value of extremeIQosProfileMinBwType.
	extremeIQosProfileMaxBwType	The type of the current maximum bandwidth setting. A value of 1 signifies that the maximum bandwidth value is a percentage of the configurable port bandwidth. A value of 2 or 3 signifies a maximum allowed bandwidth in Kbps or Mbps, respectively.

Table/Group	Supported Variables	Comments
	extremeIQosProfileMaxBw	The maximum allowed input bandwidth for this queue, expressed as either a percentage or a specific bandwidth value, as specified by the value of extremeIQosProfileMaxBwType.
	extremeIQosProfileRED	The random early drop threshold. When the input queue fill ratio exceeds this percentage, frames start to drop randomly with a linear increasing drop probability as the queue fill count approaches the maximum queue size. A value of 100 indicates that this feature is currently disabled.
	extremeIQosProfileMaxBuf	The percentage of the total ingress queue size to use. Lower values can be used to reduce the maximum latency through this queue, but with potentially greater loss with bursty traffic.
extremePerPortQosTable	extremePerPortQosIndex	The value of this variable is the same as the value of extremeQosProfileIndex of the QoS profile which is overridden (for the port specified by ifIndex) by the definition in this table.
	extremePerPortQosMinBw	The minimum percentage of bandwidth that this queue on the specified port requires. The switch is required to provide the minimum amount of bandwidth to the queue. The lowest possible value is 0%.
	extremePerPortQosMaxBw	The maximum percentage of bandwidth that this queue on the specified port is permitted to use.
	extremePerPortQosPriority	The level of priority at which this queue will be serviced by the switch.

Table/Group	Supported Variables	Comments
	extremePerPortQosRowStatus	The status of the extremePerPortQos entry. This object can be set to active (1) and createAndGo (4). The following value may be read: active (1). Note that a destroy (6) is not supported. A row will only be deleted from this table when the QoS profile indicated in that row is changed globally.
extremeQosByVlanMapping Table	extremeVlanIfIndex	Shows mapping of *VLAN* to queues for untagged packets. For tagged packets, the vpri field determines which queue the packet should be using.
	extremeQosByVlanMappingQosProfile Index	Value of extremeQosProfileIndex that uniquely identifies a QoS profile entry in an extremeQosProfileTable. This indicates the QoS to be given to traffic for this VLAN in the absence of any other more specific configuration information for this traffic.

EXTREME-PVLAN-MIB

Two MIB tables allow for GET/SET operations to view and configure ExtremeXOS PVLAN objects. The first is a PVLAN name table with PVLAN name as the index, and the second is the PVLAN subscriber table with PVLAN name, member *VLAN* type, and Subscriber VlanIfIndex as the index. Functionality includes supporting the current set of PVLAN CLI configuration options.

PVLAN Table (extremePvlanTable)

extremePvlanName (index)	extremePvlanVrName	extremePvlanNetworkVlanIfIndex
DisplayString	DisplayString	InterfaceIndexOrZero
p1	VR-Default14	14
foo	VR-Cust1	0

PVLAN Subscriber Table (extremePvlanSubscriberTable)

extremePvlanName (index)	extremePvlanSubscriberVlanIfIndex (index)	extremePvlanSubscriberType	extremePvlanSubscriberLoopBackPortIfIndex
DisplayString	InterfaceIndex	Integer	InterfaceIndexOrZero
foo	9	PVLAN_ISOLATED(2)	0
foo	12	PVLAN_ISOLATED(2)	1002
p1	1	PVLAN_ISOLATED(2)	0
p1	6	PVLAN_NON_ISOLATED(1)	1004
p1	8	PVLAN_ISOLATED(2)	1010
p1	10	PVLAN_NON_ISOLATED(1)	0

RowStatus is defined as: active(1), createAndGo(4), destroy(6)

EXTREME-SNMPv3-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeTargetAddrExtTable	extremeTargetAddrExtIgnoreMPModel	The value of the snmpTargetParamsMPModel object in the snmpTargetParamsTable is ignored while determining the version of the trap/notification to be sent. When a trap-receiver is created through the CLI command 'configure snmp add trapreceiver ....', the value of this object is set to TRUE for the corresponding entry in this table.
	extremeTargetAddrExtStandardMode	When this object is set to TRUE, the management target is treated as a 'standard mode' one, in that any Extreme Networks specific extra varbindspresent in a standards-based trap/notification is not sent to this management target. Only the varbinds defined in the standard are sent.

Table/Group	Supported Variables	Comments
	extremeTargetAddrExtTrapSrcIp	This object contains the source IP address from which traps have to be sent out. If this object is assigned an IP address that does not belong to the switch, an error is thrown.
	extremeTargetAddrExtVrName	This object contains the VR name through which the *SNMP* Traps are being sent out.
extremeUsmAesCfb192 Protocol		Supported from ExtremeXOS 12.3
extremeUsmAesCfb256 Protocol		Supported from ExtremeXOS 12.3

EXTREME-STP-EXTENSIONS-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeStpDomainTable	All objects	This table contains *STP* information per STP domain.
	extremeStpDomainStpdDescription	The description associated with this STP domain.
extremeStpPortTable	All objects	This table contains port-specific information per STP domain.
extremeStpVlanPortTable	All objects	This table contains information of the ports belonging to a STP domain on a per *VLAN* basis.
extremeStpNotifications	extremeStpEdgePortLoopDetected	This is a trap that would be sent when a loop has been detected on the network login edge safeguard port (the port will be disabled).

EXTREME-STPNOTIFICATIONS-MIB

This MIB defines the following Extreme-specific *STP* Notifications trap generated by Extreme Networks devices.

Trap	Comments
extremeStpEdgePortLoopDetected	A loop has been detected on the network login edge safeguard port and the port will be disabled.

EXTREME-SYSTEM-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
	extremeSaveConfiguration	When this object is set, the device copies the contents of the configuration database to a buffer and saves it to the persistent store specified by the value of the object. The save is performed asynchronously, and the *SNMP* agent continues to respond to both gets and sets while the save is taking place. A network management application may use the extremeSaveStatus object to determine when the asynchronous save operation has completed.
	extremeSaveStatus	This object returns the status of a save operation invoked by setting the extremeSaveConfiguration object. A network management application can read this object to determine that a save operation has completed.
	extremeCurrentConfigInUse	Shows which NVRAM configuration store was used at last boot.
	extremeConfigToUseOnReboot	Controls which NVRAM configuration store will be used on next reboot.
	extremeOverTemperatureAlarm	Alarm status of overtemperature sensor in device enclosure.
	extremePrimaryPowerOperational	Not supported. Always returns True.
	extremePowerStatus	Not supported. Always returns presentOK.

Table/Group	Supported Variables	Comments
	extremePowerAlarm	Not supported. Always returns False.
	extremeRedundantPowerStatus	
	extremeRedundantPowerAlarm	Not supported. Always returns presentOK.
	extremeInputPowerVoltage	Contains the input voltage of the latest power-supply to power-on in a system with multiple power-supplies.
	extremePrimarySoftwareRev	This value indicates the revision number of the software image on the primary partition.
	extremeSecondarySoftwareRev	This value indicates the revision number of the software image on the secondary partition.
	ExtremeImageToUseOnReboot	This value indicates the partition where the software image is located and to be used on the next boot.
	extremeSystemID	This represents the system ID of a switch.
	extremeSystemBoardID	Not supported.
	extremeSystemLeftBoardID	
	extremeSystemRightBoardID	
	extremeRmonEnable	
	extremeBootROMVersion	Returns information for the current MSM only.
	extremeDot1dTpFdbTableEnable	Not supported.
	extremeHealthCheckErrorType	
	extremeHealthCheckAction	
	extremeHealthCheckMaxRetries	
	extremeCpuUtilRisingThreshold	
	extremeCpuTaskUtilPair	
	extremeCpuAggregateUtilization	
	extremeCpuUtilRisingThreshold	
	extremeAuthFailSrcAddr	
	extremeCpuTransmitPriority	
	extremeImageBooted	

Table/Group	Supported Variables	Comments
extremeFanStatusTable	extremeFanStatusEntry	Operational status of all internal cooling fans.
	extremeFanNumber	Identifier of cooling fan, numbered from the front and/or left side of device.
	extremeFanOperational	Operational status of a cooling fan.
	extremeFanEntPhysicalIndex	The entity index for this fan entity in the entityPhysicalTable table of the entity MIB.
	extremeFanSpeed	The speed (RPM) of a cooling fan in the fantray.
extremeCpuTaskTable	All objects	Not supported.
extremeCpuTask2Table	All objects	Not supported.
extremeSlotTable	All objects	Cards are currently not configurable via SNMP.
extremePowerSupplyEntPhysicalIndex	extremePowerSupplyEntPhysicalIndex	The entity index for this psu entity in the entityPhysicalTable of the entity MIB.
extremePowerSupplyNumber	extremePowerSupplyNumber	Power supply number.
extremePowerSupplySerialNumber	extremePowerSupplySerialNumber	The serial number of the power supply unit.
extremePowerSupplySource	extremePowerSupplySource	The power supply unit input source.
extremePowerSupplyTable	extremePowerSupplyStatus	Status of the power supply.
	extremePowerSupplyInputVoltage	Input voltage of the power supply.
	extremePowerSupplyFan1Speed	The speed of Fan-1 in the power supply unit.
	extremePowerSupplyFan2Speed	The speed of Fan-2 in the power supply unit.
	extremePowerSupplyInputPowerUsage	Input power usage for the given psu slot. The value 0 in this field indicates the power usage is not supported or that there is a read failure.

Table/Group	Supported Variables	Comments
	extremePowerMonSupplyNumOutput	Number of output sensors in the power supply unit.
	extremePowerSupplyInputPowerUsageUnitMultiplier	The magnitude of watts for the usage value in extremePowerSupplyInputPowerUsage.
	extremePowerSupplyTotalPowerAvailable	Total power available.
	extremePowerSupplyOutputVoltage	Output voltage.
	extremePowerSupplyOutputCurrent	Output current.
extremePowerSupplyOutputPowerTable	extremePowerSupplyIndex	Power supply unit slot index.
	extremePowerSupplyOutputSensorIdx	Power supply Sensor index.
	extremePowerSupplyOutputVoltage	Output voltage per sensor for the current PSU slot number. A zero (0) in this field indicates that the PSU does not support output voltage reading or else there is an output voltage read error.
	extremePowerSupplyOutputCurrent	Output current per sensor for the current PSU slot number. A zero (0) in this field indicates that the PSU does not support output current reading or else there is an output current read error.
	extremePowerSupplyOutputUnitMultiplier	The magnitude of volts and amps for the usage value in extremePowerSupplyOutputVoltage and extremePowerSupplyOutputCurrent.
extremePowerSupplyUsageTable	extremeSlotIndex	Slot number in the stack- based system.

Table/Group	Supported Variables	Comments
	extremePowerSupplyUsageValue	Power usage of the particular slot in the stack. The power usage is measured in milliwatts.
	extremePowerSupplyUnitMultiplier	The magnitude of watts for the usage value in extremePowerSupplyUsageValue.
extremeSystemPowerUsage	extremeSystemPowerUsageValue	The current power usage of the system. In stack mode. this variable tells the total power usage of the entire system.
	extremeSystemPowerUsageUnitMultiplier	The magnitude of watts for the usage value in extremeSystemPowerUsageValue.
extremeSystemPowerMonitorTable	extremeSystemPowerMonitorIndex1	Reserved. Can be used for future expansion. Currently set to zero.
	extremeSystemPowerMonitorPollInterval	Configure how often input power is measured. It is configured in seconds with a default value of 60 seconds. If zero (0) is configured, then the input power measurement is disabled.
	extremeSystemPowerMonitorReportChanges	Configure report-changes. Has none or log or trap or log-and-trap, with a default of none.
	extremeSystemPowerMonitorChangeThreshold	Configure the input power change threshold to initiate report-changes action. By default 2 watts is configured. This field is configured in watts.

Table/Group	Supported Variables	Comments
extremeSystemPowerUsageNotification	sysUpTime, sysDescr, extremeSystemPowerUsageValue, extremeSystemPowerUsageUnitMultiplier	Whenever the power usage is increased/ decreased by the configured threshold value, then the power usage trap is generated if the trap is enabled.
extremeImageTable	extremeImageNumber	This table contains image information for all images installed on the device.extremeImageNumber values are not compatible with EW releases. Current image has value 3 instead of 0.
	extremeMajorVersion	This is the first number within the software image version number which consists of 4 numbers separated by a period.
	extremeMinorVersion	This is the second number within the software image version number.
	extremeBuildNumber	This is the fourth number within the software image version number.
	extremeTechnologyReleaseNumber	The technology release version. This value is zero for all but TR releases.
	extremeSustainingReleaseNumber	The sustaining release number for the ExtremeXOS version.
	extremeBranchRevisionNumber	This is the branch from where the software image was built.
	extremeImageType	This is the software image type (for example, ExtremeXOS core, ExtremeXOS module, ExtremeXOS firmware).

Table/Group	Supported Variables	Comments
	extremeImageDescription	Description of image contains image version, including major version, submajor version, minor version, build version, build branch, build-master login, and build date.
	extremePatchVersion	The ExtremeXOS release patch version. This is the third number within the software image version number.
extremeImageFeatureTable	extremeImageFeatureEntry	A table containing information about the software features.
	extremeImageFeatureNumber	A unique integer identifying the particular software image. This indicates the partition on which the image is loaded: 0—Current partition1—Primary partition2—Secondary partition
	extremeImageSshCapability	Indicates whether image has SSH capability: 1=nossh—shown by default when the SSH license is not enabled.2=ssh—shown when the SSH license is enabled.
	extremeImageUAACapability	Not supported.
extremeCpuMonitorInterval		This value determines how frequent CPU usage will be monitored.
extremeCpuMonitorTotalUtilization		This value indicates the total CPU utilization.
extremeCpuMonitorTable		This value indicates the total CPU utilization.
	extremeCpuMonitorSlotId	This value indicates the slot ID where the CPU is being monitored.

Table/Group	Supported Variables	Comments
	extremeCpuMonitorProcessName	This value indicates the process name for the process being monitored.
	extremeCpuMonitorProcessId	This value indicates the process ID.
	extremeCpuMonitorProcessState	This value indicates the current state of the process.
	extremeCpuMonitorUtilization5secs	This value indicates the CPU utilization in the past 5 seconds.
	extremeCpuMonitorUtilization10secs	This value indicates the CPU utilization in the past 10 seconds.
	extremeCpuMonitorUtilization30secs	This value indicates the CPU utilization in the past 30 seconds.
	extremeCpuMonitorUtilization1min	This value indicates the CPU utilization in the past 1minute.
	extremeCpuMonitorUtilization5mins	This value indicates the CPU utilization in the past 5 minutes.
	extremeCpuMonitorUtilization30mins	This value indicates the CPU utilization in the past 30 minutes.
	extremeCpuMonitorUtilization1hour	This value indicates the CPU utilization in the past 1 hour.
	extremeCpuMonitorUserTime	This value indicates the CPU usage under User Mode.
	extremeCpuMonitorSystemTime	This value indicates the CPU usage under system mode.
extremeCpuMonitorSystemTable		This table contains CPU monitoring information for all system processes.
	extremeCpuMonitorSystemSlotId	This value indicates the slot ID where the CPU is being monitored.
	extremeCpuMonitorSystemUtilization5secs	This value indicates the CPU utilization in the past 5 seconds.

Table/Group	Supported Variables	Comments
	extremeCpuMonitorSystemUtilization10secs	This value indicates the CPU utilization in the past 10 seconds.
	extremeCpuMonitorSystemUtilization30secs	This value indicates the CPU utilization in the past 30 seconds.
	extremeCpuMonitorSystemUtilization1min	This value indicates the CPU utilization in the past 1 minute.
	extremeCpuMonitorSystemUtilization5mins	This value indicates the CPU utilization in the past 5 minutes.
	extremeCpuMonitorSystemUtilization30mins	This value indicates the CPU utilization in the past 30 minutes.
	extremeCpuMonitorSystemUtilization1hour	This value indicates the CPU utilization in the past 1 hour.
	extremeCpuMonitorSystemMaxUtilization	This value indicates the maximum CPU utilization so far.
extremeMemoryMonitorSystem Table		This table contains system-level memory monitor information.
	extremeMemoryMonitorSystemSlotId	This value indicates the slot ID where the memory is being monitored.
	extremeMemoryMonitorSystemTotal	This value indicates the total memory installed on the switch.
	extremeMemoryMonitorSystemFree	This value indicates the amount of memory that is free.
	extremeMemoryMonitorSystemUsage	This value indicates the amount of memory consumed for kernel code.
	extremeMemoryMonitorUserUsage	This value indicates the amount of memory consumed by user processes as well as the kernel.
extremeMemoryMonitorTable		This table contains memory monitor information for each user process.

Table/Group	Supported Variables	Comments
	extremeMemoryMonitorSlotId	This value indicates the slot ID where the memory is being monitored.
	extremeMemoryMonitorProcessName	This value indicates the name of the process being monitored.
	extremeMemoryMonitorUsage	This value indicates the amount of memory being consumed by this user process.
	extremeMemoryMonitorLimit	Not supported.
	extremeMemoryMonitorZone	
	extremeMemoryMonitorGreenZoneCount	
	extremeMemoryMonitorYellowZoneCount	
	extremeMemoryMonitorOrangeZoneCount	
	extremeMemoryMonitorRedZoneCount	
	extremeMemoryMonitorGreenZoneThreshold	
	extremeMemoryMonitorYellowZoneThreshold	
	extremeMemoryMonitorOrangeZoneThreshold	
	extremeMemoryMonitorRedZoneThreshold	

EXTREME-TRAP-MIB

This MIB defines the following Extreme-specific SNMPv1 traps generated by Extreme Networks devices.

Trap	Comments
extremeOverheat	The on-board temperature sensor has reported an overheat condition. The system shuts down until the unit has sufficiently cooled such that operation can begin again. A cold start trap is issued when the unit has come back on line.
extremeFanfailed	One or more of the cooling fans inside the device has failed. A fanOK trap will be sent once the fan has attained normal operation.

Trap	Comments
extremeFanOK	A fan has transitioned out of a failure state and is now operating correctly.
extremeInvalidLoginAttempt	A user attempted to log into the console or by telnet but was refused access due to an incorrect username or password.
extremePowerSupplyGood	One or more previously bad sources of power to this agent has come back to life without causing an agent restart.
extremePowerSupplyFail	One or more sources of power to this agent has failed. Presumably a redundant power-supply has taken over.
extremeEdpNeighborAdded	This node discovers a new neighbor through *EDP*.
extremeEdpNeighborRemoved	No EDP updates are received from this neighbor within the configured timeout period and this neighbor entry is aged out by the device.
extremeModuleStateChanged	Signifies that the value of the extremeSlotModuleState for the specified extremeSlotNumber has changed. Traps are reported only for significant states.

EXTREME-V2TRAP-MIB

This MIB defines the following Extreme-specific SNMPv2c traps generated by Extreme Networks devices.

Trap	Comments
extremeHealthCheckFailed	CPU HealthCheck has failed.
extremeBgpM2PrefixReachedThreshold	This notification is generated when the number of prefixes received over this peer session reaches the threshold limit.
extremeBgpM2PrefixMaxExceeded	This notification is generated when the number of prefixes received over this peer session reaches the maximum configured limit.
extremeEapsStateChange	Send on master/transit nodes.
extremeEapsFailTimerExpFlagSet	This notification is generated when the *EAPS* domain fail timer expires for the first time, while its state is not in Fail state.
extremeEapsFailTimerExpFlagClear	This notification is generated when the EAPS domain fail timer expired flag is cleared.
extremeEapsLinkDownRingComplete	If a transit is in link-down state, and it receives a Health-Check-Pdu from the master indicating the ring is complete, there is some problem with the transit switch that has issued this trap message.
extremeEapsLastStatusChangeTime	Sent on master/transit nodes. Provides a general indication of a status change using a 10 second timer.

Trap	Comments
extremeEapsPortStatusChange	Sent on master/transit nodes.
extremeEapsConfigChange	Sent on master/transit nodes. This trap has a granularity of 30 seconds.
extremeEapsSharedPortStateChange	Sent on controller/partner nodes.
extremeEapsRootBlockerStatusChange	
extremeOverheatNormal	An overheat (return to) normal notification indicates that the on-board tempature sensor has reported a temperature that has returned to within the normal operating range from having been in an overheat (or overcold) condition. The temperature of the unit has sufficiently cooled to be below the maximum (or warmed to be above the minimum) of the normal operating range.

EXTREME-VLAN-MIB

The following tables, groups, and variables are supported in this MIB.

Note
SNMP Set operations are not allowed for "Mgmt" and "Default" VLANs.

Table/Group	Supported Variables	Comments
extremeVirtualGroup	extremeNextAvailableVirtIfIndex	
extremeVlanIfTable	extremeVlanIfIndex	While creating a new row in the extremeVlanIfTable, the value of the object extremeVlanIfDescr must be specified. For all tables in this MIB that contain objects with RowStatus semantics, the only values supported are: {active, createAndGo, destroy}.
	extremeVlanIfDescr	This is a description of the VLAN interface.
	extremeVlanIfType	The VLAN interface type.
	extremeVlanIfGlobalIdentifier	Not supported.

Table/Group	Supported Variables	Comments
	extremeVlanIfStatus	The status column for this VLAN interface. This object can be set to: active (1); createAndGo (4); createAndWait (5); destroy (6). The following values may be read: active (1); notInService (2); notReady (3).
	extremeVlanIfIgnoreStpFlag	Not supported.
	extremeVlanIfIgnoreBpduFlag	Not supported.
	extremeVlanIfLoopbackModeFlag	Setting this object to true causes loopback mode to be enabled on this VLAN.
	extremeVlanIfVlanId	The VLAN ID of this VLAN.
extremeGlobalMappingTable	Not supported	
extremeVlanEncapsTable	Not supported	
extremeVlanIpTable	All objects	For all tables in this MIB that contain objects with RowStatus semantics, the only values supported are: {active, createAndGo, and destroy}.
extremeVlanProtocolTable	Not supported	
extremeVlanProtocolBindingTable	Partial	For all tables in this MIB that contain objects with RowStatus semantics, the only values supported are: {active, createAndGo, and destroy}. New to the ExtremeXOS software: association of a protocol filter and VLAN.
extremeVlanProtocolVlanTable	Not supported	
extremeVlanProtocolDefTable	Partial	For all tables in this MIB that contain objects with RowStatus semantics, the only values supported are: { active, createAndGo, and destroy} New to the ExtremeXOS software: This is a new table introduced to add a protocol filter with etype values during creation of the filter itself.

Table/Group	Supported Variables	Comments
extremeVlanOpaqueTable	All objects	This is a read only table. For adding ports to a VLAN or deleting ports from a VLAN, use extremeVlanOpaqueControlTable.
extremeVlanOpaqueControlTable	All objects	For all tables in this MIB that contain objects with RowStatus semantics, the only values supported are: {active, createAndGo, and destroy}. New to the ExtremeXOS software: extremeVlanOpaqueControlTable is a write only table and cannot be used to read. This is used to add/delete ports on a VLAN.
extremeVlanStackTable	All objects	Not supported.
extremeVlanL2StatsTable	All objects	Not supported. This table contains per VLAN information about the number of packets sent to the CPU, the number of packets learned, the number of *IGMP (Internet Group Management Protocol)* control packets snooped and the number of IGMP data packets switched. This is the same information that is available using the CLI command show l2stats.
extremePortVlanStatsTable		VLAN statistics per port.
extremePortVlanStatsEntry	extremeStatsPortIfIndex	The index of this table.
	extremeStatsVlanNameIndex	The index of this table.
	extremePortVlanStatsCntrType	Read-only.
	extremePortVlanTotalReceived BytesCounter	The total number of bytes received by a port for a particular VLAN.
	extremePortVlanTotalReceived FramesCounter	The total number of frames received by a port for a particular VLAN.
	extremePortVlanTotalTransmittedBytesCounter	The total number of bytes transmitted by a port for a particular VLAN.

Table/Group	Supported Variables	Comments
	extremePortVlanTotalTransmittedFramesCounter	The total number of frames transmitted by a port for a particular VLAN.
	extremePortConfigureVlanStatus	The row status variable, used according to row installation and removal conventions.

EXTREME-VM-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
extremeVMFTPServer Table	extremeVMFTPServerEntry	
	extremeVMFTPServerType	The type of the FTP server. The backup server is contacted if the primary fails to respond.
	extremeVMFTPAddrType	The IP type of IP address.
	extremeVMFTPServer	The IP address of the FTP server used for transferring various management files.
	extremeVMFTPSynchInterval	The time in minutes between automatic synchronization attempts. A value of 0 indicates that automatic synchronizations are not performed. Note that each switch does not perform a synchronization at exactly the time configured, but varies the synchronization interval between 3/4 and 5/4 of the configured interval. This avoids the situation where all switches in a network attempt a synchronization at exactly the same moment. Automatic synchronization is disabled by default, and requests to enable them are rejected until the FTP server information (IPv4 or IPv6 address, username, and password) is configured.
	extremeVMFTPRowStatus	There can only be two entries in this table, one each for primary and secondary FTP servers.

Table/Group	Supported Variables	Comments
	extremeVMFTPPathName	The FTP server directory name for the policies to be synchronized. A value of '/pub' is used by default.
	extremeVMFTPUsername	A valid username on the FTP server.
	extremeVMFTPPassword	The password associated with the FTP user. This object returns a zero length string when queried.
extremeVMGeneral	extremeVMFTPPolicyDir	The server directory name for the policies to be synchronized. A value of '/' is used by default.
	extremeVMLastSynch	The timestamp of the most recent synchronization attempt.
	extremeVMLastSynchStatus	The result of the most recent synchronization attempt:success (1)—indicates that the synchronization completed successfullyaccessDenied (2)—The username and password were not accepted by the serverserverTimeout (3)—Could not establish a file transfer session with the configured serverserverNotConfigured (4)—The server configuration is not complete.
	extremeVMSynchAdminState	Triggers a synchronization cycle on demand. A synchronization automatically downloads new or updated policies as well as delete policies to match those on the server: idle (1) is returned whenever this object is read.synchronizeNow (2) triggers an immediate synchronization, and is reflected in extremeVMSynchOperState. Attempts to set this variable to synchronizeNow (2) are rejected if a synchronization is currently in progress.
	extremeVMSynchOperState	Indicates if a synchronization is in progress, either on-demand or automatic.

Table/Group	Supported Variables	Comments
	extremeVMTrackingEnabled	The virtual machine (VM) tracking feature is disabled by default and can be enabled using this object.
	extremeVMPortConfigTable	Configures the VM features on each port.
extremeVMPortConfig Entry		An entry in the table for VM features on each port.
	extremeVMPortConfigIfIndex	The value of ifIndex of a physical port capable of supporting the VM tracking features.
	extremeVMPortConfigVMTracking Enabled	Enables the VM tracking feature on a port. The VM tracking feature is disabled by default.
extremeVMVPPTable		This table contains port policies contained within this switch. Port policies come in two variants: network and local. Network policies are downloaded from the FTP server; local policies reside only within a single switch.
extremeVMVPPEntry		An entry in the table of VM policy information of this device. This table is populated with two sets of policies, those downloaded from the policy server and those defined locally on this switch.
	extremeVMVPPType	The type of the port policy. Network port policies are obtained from a central policy store. Local policies are specific to this particular switch.
	extremeVMVPPName	The name of the port VPP. VPP names must be alpha-numeric and must start with an alpha character.
	extremeVMVPPControl	Performs the requested operation on this policy. synchronizeNow (1) downloads a copy of the policy from the FTP server. (Network policies only). This object returns noOperation (2) if read.
	extremeVMVPPRowStatus	Only local VPPs can be created or deleted.

Table/Group	Supported Variables	Comments
extremeVMMappingTable		This table contains the mapping of port policies to VM MAC addresses.
	extremeVMMappingEntry	An entry in the table of VM information of this device.
	extremeVMMappingType	The type of mapping for this entry. A local mapping exists only on this specific switch. A network mapping is one obtained through a download of a mapping file.
	extremeVMMappingMAC	The MAC address associated with the VM. Note that a VM can have multiple MAC addresses.
	extremeVMMappingIngressVPPName	The ingress policy associated with the VM/MAC address. Note that this may refer to a policy without a corresponding entry in the extremeVMVPPTable if a network policy mapping refers to a non-existent policy. This indicates an error in the policy mapping file that is consulted if network authentication fails. When creating an entry in this table, this name must refer to an existing, valid, local policy. The creation of a mapping to a network policy is not permitted. Those mappings must be created at the central policy server.
	extremeVMMappingEgressVPPName	The egress policy associated with the VM/MAC address. Note that this might refer to a policy without a corresponding entry in the extremeVMVPPTable if a network policy mapping refers to a non-existent policy. This indicates an error in the policy mapping file that is consulted if network authentication fails. When creating an entry in this table, this name must refer to an existing, valid, local policy. The creation of a mapping to a network policy is not permitted. Those mappings must be created at the central policy server.

Table/Group	Supported Variables	Comments
	extremeVMMappingStatus	Indicates the virtual port profile mapping status: vppValid (1)—A VPP mapped to this VM MAC address does not have any policies associated with it (or) all the policies associated with this VPP can be applied (policy validation is success). Policy validation will happen only when this VM MAC is detected.vppMissing (2)—This value is applicable only for network VM if the specified VPP name was missing.vppInvalid (3)—One of the polices mapped to VPP cannot be applied (policy validation failed) because the policy file contains one or more errors that prevent it from being appliedvppNotMaped (4)—The VM does not have any VPP mapped.
	extremeVMMappingRowStatus	Only local polices can be created or deleted.
		The virtual port profile associated with the VM MAC address. When creating an entry in this table, this name must refer to an existing, valid, local profile. The creation of a mapping to a network profile is not permitted. Those mappings must be created at the central policy server.
extremeVMVPP2PolicyTable		This table contains the mapping of a VPP to individual policies.
extremeVMVPP2PolicyEntry		An individual mapping of VPP to policy.
	extremeVMVPP2PolicyVPPName	The name of the VPP.
	extremeVMVPP2PolicyPolicyName	The name of the local policy.
	extremeVMVPP2PolicyType	The type of policy
	extremeVMVPP2PolicyOrder	The order in which this policy is executed.
	extremeVMVPP2PolicyRowStatus	The row status for this mapping.

Table/Group	Supported Variables	Comments
extremeVMDetected	extremeVMDetectedNumber	The number of VMs detected on this switch.
extremeVMDetectedTable		This table contains the currently detected VMs on this switch.
extremeVMDetectedEntry		An entry in the table of VM information of this device.
	extremeVMDetectedMAC	The MAC address associated with the VM. Note that a VM can have multiple MAC addresses.
	extremeVMDetectedVMName	The name of the VM. Note that a VM authenticated locally might not have a name.
	extremeVMDetectedIngressVPPName	The name of the policy applied (or attempted to apply) to this VM.
	extremeVMDetectedEgressVPPName	The name of the policy applied (or attempted to apply) to this VM.
	extremeVMDetectedIfIndex	The value of ifIndex on which this VM was detected.
	extremeVMDetectedAdminStatus	The administrative status of the VM authentication. Setting this variable to authenticating (1) will force the re-authentication of the VM. This variable always returns idle (2) when read.
	extremeVMDetectedOperStatus	The authentication status of the VM: authenticating (1)—an authentication is currently in progressauthenticatedNetwork (2)—the VM has been authenticated by a network sourceauthenticatedLocally (3)—the VM has been authenticated by the local databasedenied (4)—the VM was explicitly denied entrynotAuthenticated (5)—the authentication process timed out or was never attempted

Table/Group	Supported Variables	Comments
	extremeVMDetectedResultIngress	Indicates the result of a VM entry into the network. Only the two values below will be returned at any point of time: policyApplied (1)—All the ingress policies in the VPP were successfully applied to the port.policyNotApplied (2) - One of the ingress policies in the VPP was not applied to the port. If this value is returned then refer to the extremeVMDetectedIngErrPolicies object for list of failed ingress policies.
	extremeVMDetectedResultEgress	Indicates the result of a VM entry into the network. Only the two values below will be returned at any point of time: policyApplied (1) - All the egress policies in the VPP were successfully applied to the port.policyNotApplied (2) - One of the egress policies in the VPP was not applied to the port. If this value is returned then refer to the extremeVMDetectedEgrErrPolicies object for list of failed egress policies.
	extremeVMDetectedIngErrPolicies	Displays the list of failed ingress policies.
	extremeVMDetectedEgrErrPolicies	Displays the list of failed egress policies.
	extremeVMDetectedVPPName	The name of the VPP applied (or attempted to apply) to this VM.
	extremeVMDetectedVPPResult	Indicates the result of a VPP associated with a VM MAC: vppMapped (1)—indicates that the named VPP was mapped.vppNotMapped (2)—indicates that the no VPP was mapped.vppInvalid (3)—indicates that the VPP mapped was invalid.vppMissing (4)—indicates that the VPP mapped was missing
extremeVMVPPDetailTable		This table contains the mapping of a VPP to individual policies.

Table/Group	Supported Variables	Comments
extremeVMVPPDetailEntry		A set of mappings from a VPP to one or more policies.
	extremeVMVPPDetailVPPName	The name of the VPP.
	extremeVMVPPDetailDirection	The direction in which the policy is applied.
	extremeVMVPPDetailType	The type of policy.
	extremeVMVPPDetailOrder	The order in which this policy is executed.
	extremeVMVPPDetailPolicyName	The name of the local policy
	extremeVMVPPDetailRowStatus	The row status for this mapping.
extremeVMVPPSynchType		Indicates the type of policy

The following traps can be generated.

Trap	Notification Objects	Comments
extremeVMNotificationPrefix	extremeVMVPPSyncFailed	A synchronization attempt failed.
	extremeVMVPPInvalid	A VPP definition is invalid, indicating it cannot be applied to a port.
	extremeVMMapped	This notification is generated whenever a MAC address is manually mapped to a local policy.
	extremeVMUnMapped	This notification is generated whenever a MAC address is manually unmapped to a local policy.
	extremeVMDetectResult	This notification is generated after a VM is detected on a port and reflects the result of that operation.
	extremeVMUnDetectResult	This notification is generated after a VM is undetected (removed) from a port.

MIB Support Details

The following sections describe the MIB support provided by the ExtremeXOS *SNMP* agent residing on Extreme Networks devices running ExtremeXOS:

Where applicable, these sections note how the implementation differs from the standards, or from the private MIBs.

Note

Only entries for the default VR are supported.

ExtremeXOS software supports only the following “Row status” in the MIBs:

- Create and Go
- Active
- Destroy

The standard MIBs are described in the following sections.

ETMONSERVICEABILITYTRAP

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
etmonServiceabilityTrapPrefix	noMoreMemory	Status = current. "Memory full, No more memory. System memory is depleted."
	asyncQGrowing	Status = current. "async queue is growing. Logged messages in switch will provide more information related to growing async queue. MIB have limitation to provide all the information related to growing sync queue."

IEEE 802.1AB (LLDP-MIB)

All tables and variables of this MIB are supported.

IEEE 802.1AB (LLDP-EXT-DOT1-MIB)

All tables and variables of this MIB are supported.

IEEE 802.1AB (LLDP-EXT-DOT3-MIB)

All tables and variables of this MIB are supported.

IEEE 802.1AG (CFM MIB)

This MIB contains objects for the 802.1ag protocol.

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
dot1agCfmMdTable	dot1agCfmMdIndex	The index to the Maintenance Domain (MD) table.
	dot1agCfmMdFormat	Supported as read only. The type (and thereby format) of the MD name.
	dot1agCfmMdName	Supported as read only. The MD name.
	dot1agCfmMdMdLevel	Supported as read only. The MD Level.
	dot1agCfmMdMhfCreation	Enumerated value indicating whether the management entity can create MHFs (MIP Half Function) for this MD. Supported as read only. Configuration in CLI is not supported. The defMHFdefault (2) value is returned.
	dot1agCfmMdMhfIdPermission	Supported as read only. Configuration in CLI is not supported. The sendIdChassisManage (4) value is returned.
	dot1agCfmMdMaNextIndex	Value to be used as the index of the Maintenance Association (MA) table entries.
	dot1agCfmMdRowStatus	The status of the row. The writable columns in a row cannot be changed if the row is active. All columns must have a valid value before a row can be activated.
dot1agCfmMdTableNextIndex	dot1agCfmMdTableNextIndex	An unused value for dot1agCfmMdIndex that is used as the MD index for the next newly created domain.
dot1agCfmMaNetTable	dot1agCfmMdIndex	MD index.
	dot1agCfmMaIndex	Index of the MA table. dot1agCfmMdMaNextIndex needs to be inspected to find an available index for row-creation.

Table/Group	Supported Variables	Comments
	dot1agCfmMaNetFormat	Supported as read only. The type of the MA name.
	dot1agCfmMaNetName	Supported as read only. MA name.
	dot1agCfmMaNetCcmInterval	Supported as read only. Transmission interval between CCMs to be used by all MEPs in the MA.
	dot1agCfmMaNetRowStatus	The status of the row. The writable columns in a row cannot be changed if the row is active. All columns must have a valid value before a row can be activated.
dot1agCfmMepTable	dot1agCfmMdIndex	MD Index.
	dot1agCfmMaIndex	MA Index.
	dot1agCfmMepIdentifier	A small integer, unique over a given MA identifying a specific MA end point.
	dot1agCfmMepIfIndex	Supported as read only. Interface index of the interface bridge port.
	dot1agCfmMepDirection	Supported as read only. The direction in which the MEP faces on the bridge port.
	dot1agCfmMepPrimaryVid	Supported as read only. VID of the MEP.
	dot1agCfmMepActive	Supported as read only. Administrative state of the MEP.
	dot1agCfmMepFngState	Current state of the MEP Fault Notification Generator State Machine. Can have any one of the following values:fngReset (1)fngDefect (2)fngReportDefect (3)fngDefectReported (4)fngDefectClearing (5)
	dot1agCfmMepCciEnabled	Supported as read only. If set to true, the MEP will generate CCM.

Table/Group	Supported Variables	Comments
	dot1agCfmMepCcmLtmPriority	Supported as read only. The priority value for CCMs and LTMs transmitted by the MEP. Configuration in CLI is not supported. Default value 0 is returned.
	dot1agCfmMepMacAddress	MAC address of the MEP. In Extreme Networks switches, the MAC is returned.
	dot1agCfmMepLowPrDef	Supported as read only. An integer value specifying the lowest priority defect that is allowed to generate a Fault Alarm. Configuration in CLI is not supported. Default VLAN macRemErrXon is returned.
	dot1agCfmMepFngAlarmTime	Supported as read only. The time that defects must be present before a Fault Alarm is issued. Configuration in CLI is not supported. Default value 2.5 seconds is returned.
	dot1agCfmMepFngResetTime	Supported as read only. The time that defects must be absent before resetting a Fault Alarm. Configuration in CLI is not supported. Default value 10 seconds is returned.
	dot1agCfmMepHighestPrDefect	The highest priority defect that has been present. Possible values are:none (0)defRDICCM (1)defMACstatus (2)defRemoteCCM (3)defErrorCCM (4)defXconCCM (5)
	dot1agCfmMepDefects	Error condition to be sent. The conditions can be any one of the following:bDefRDICCM (0)bDefMACstatus (1)bDefRemoteCCM (2)bDefErrorCCM (3)bDefXconCCM (4)

Table/Group	Supported Variables	Comments
	dot1agCfmMepErrorCcmLastFailure	The last-received CCM that triggered a DefErrorCCM fault.
	dot1agCfmMepXconCcmLastFailure	The last-received CCM that triggered a DefXconCCM fault.
	dot1agCfmMepCcmSequenceErrors	The total number of out-of-sequence CCMs received from all remote MEPs.
	dot1agCfmMepCciSentCcms	The total number of Continuity Check messages transmitted.
	dot1agCfmMepNextLbmTransId	Next sequence number/ transaction identifier to be sent in a Loopback message.
	dot1agCfmMepLbrIn	The total number of valid, in-order Loopback Replies received.
	dot1agCfmMepLbrInOutOfOrder	The total number of valid, out-of-order Loopback Replies received.
	dot1agCfmMepLbrBadMsdu	The total number of LBRs received whose mac_service_data_unit did not match (except for the OpCode) that of the corresponding LBM.
	dot1agCfmMepLtmNextSeqNumber	Next transaction identifier/sequence number to be sent in a Linktrace message.
	dot1agCfmMepUnexpLtrIn	The total number of unexpected LTRs received.
	dot1agCfmMepLbrOut	The total number of Loopback Replies transmitted.
	dot1agCfmMepTransmitLbmStatus	Supported as read only. A Boolean flag set to true by the bridge port to indicate that another LBM may be transmitted.
	dot1agCfmMepTransmitLbmDestMac Address	Supported as read only. The target MAC address field to be transmitted.

Table/Group	Supported Variables	Comments
	dot1agCfmMepTransmitLbmDestMepId	Supported as read only. To transmit the LBM, destMEPID need not be given.
	dot1agCfmMepTransmitLbmDestIsMepId	Supported as read only. This always returns FALSE.
	dot1agCfmMepTransmitLbmMessages	Supported as read only. The number of Loopback messages to be transmitted.
	dot1agCfmMepTransmitLbmDataTlv	Supported as read only. An arbitrary amount of data to be included in the Data TLV if the Data TLV is selected to be sent. This returns 0.
	dot1agCfmMepTransmitLbmVlanPriority	Priority. A three-bit value to be used in the VLAN tag, if present in the transmitted frame. Configuration in CLI is not supported. Default value 0 is returned.
	dot1agCfmMepTransmitLbmVlanDropEnable	Not supported. Returns FALSE.
	dot1agCfmMepTransmitLbmResultOK	Indicates the result of the operation linked with MEP active state.
	dot1agCfmMepTransmitLbmSeqNumber	The Loopback Transaction Identifier of the first LBM (to be) sent. The value returned is undefined if dot1agCfmMepTransmitLbmResultOK is false.
	dot1agCfmMepTransmitLtmStatus	Supported as read only. A Boolean flag set to true by the bridge port to indicate that another LTM may be transmitted.
	dot1agCfmMepTransmitLtmFlags	Supported as read only. The flags field for LTMs transmitted by the MEP. Currently, useFDBonly (0) is supported.
	dot1agCfmMepTransmitLtmTargetMacAddress	Supported as read only. The target MAC address field to be transmitted.

Table/Group	Supported Variables	Comments
	dot1agCfmMepTransmitLtmTargetMepId	Not supported. To transmit the LTM destMEPID need not be given. Value 0 is returned.
	dot1agCfmMepTransmitLtmTargetIsMepId	Not supported. This always returns FALSE.
	dot1agCfmMepTransmitLtmTtl	Supported as read only. The LTM TTL field.
	dot1agCfmMepTransmitLtmResult	Supported as read only. Indicates the result of the operation. Linked with MEP active state.
	dot1agCfmMepTransmitLtmSeqNumber	The LTM transaction identifier.
	dot1agCfmMepTransmitLtmEgressIdentifier	Supported as read only. Identifies the MEP Linktrace Initiator that is originating or the Linktrace Responder that is forwarding this LTM.
	dot1agCfmMepRowStatus	The status of the row. The writable columns in a row cannot be changed if the row is active. All columns must have a valid value before a row can be activated.
dot1agCfmMepDbTable	dot1agCfmMdIndex	MD Index.
	dot1agCfmMaIndex	MA Index.
	dot1agCfmMepIdentifier	MA end point identifier.
	dot1agCfmMepDbRMepIdentifier	Maintenance association end point identifier of a remote MEP whose information from the MEP database is to be returned.
	dot1agCfmMepDbRMepState	The operational state of the remote MEP IFF state machines. The state can be any one of the following:rMepIdle (1)rMepStart (2)rMepFailed (3)rMepOk (4)

Table/Group	Supported Variables	Comments
	dot1agCfmMepDbRMepFailedOkTime	The time (SysUpTime) at which the IFF remote MEP state machine last entered either the RMEP_FAILED or RMEP_OK state.
	dot1agCfmMepDbMacAddress	The MAC address of the remote MEP.
	dot1agCfmMepDbRdi	State of the RDI bit in the last received CCM.
	dot1agCfmMepDbPortStatusTlv	An enumerated value of the Port status TLV received in the last CCM from the remote MEP or the default value. The value is one of the following:psNoPortStateTLV (0)psBlocked (1)psUp (2)
	dot1agCfmMepDbInterfaceStatusTlv	An enumerated value of the Interface status TLV received in the last CCM from the remote MEP. The value can be one of the following:isNoInterfaceStatusTLV (0)isUp (1)isDown (2)isTesting (3)isUnknown (4)isDormant (5)isNotPresent (6)isLowerLayerDown (7)
	dot1agCfmMepDbChassisIdSubtype	networkAddress (5) is returned if senderIDTLV is received.
	dot1agCfmMepDbChassisId	The first octet contains the IANA Address Family Numbers enumeration value for the specific address type, and octets 2 through N contain the network address value in network byte order.
	dot1agCfmMepDbManAddressDomain	Not supported. Value zero is returned.
	dot1agCfmMepDbManAddress	Not supported. Value zero is returned.
dot1agCfmLtrTable	dot1agCfmMdIndex	MD Index.
	dot1agCfmMaIndex	MA Index.
	dot1agCfmMepIdentifier	MA end point identifier.

Table/Group	Supported Variables	Comments
	dot1agCfmLtrSeqNumber	Transaction identifier/ sequence number returned by a previous transmit linktrace message command.
	dot1agCfmLtrReceiveOrder	An index to distinguish among multiple LTRs with the same LTR Transaction Identifier field value.
	dot1agCfmLtrTtl	TTL field value for a returned LTR.
	dot1agCfmLtrForwarded	Indicates if a LTM was forwarded by the responding MP.
	dot1agCfmLtrTerminalMep	Not supported. Value FALSE is returned.
	dot1agCfmLtrLastEgressIdentifier	An octet field holding the Last Egress Identifier returned in the LTR Egress Identifier TLV of the LTR.
	dot1agCfmLtrNextEgressIdentifier	An octet field holding the Next Egress Identifier returned in the LTR Egress Identifier TLV of the LTR.
	dot1agCfmLtrRelay	Value returned in the Relay Action field.
	dot1agCfmLtrChassisIdSubtype	networkAddress (5) is returned if senderIDTLV is received.
	dot1agCfmLtrChassisId	The first octet contains the IANA Address Family Numbers enumeration value for the specific address type, and octets 2 through N contain the network address value in network byte order.
	dot1agCfmLtrManAddressDomain	Not supported. Value zero is returned.
	dot1agCfmLtrManAddress	Not supported. Value zero is returned.
	dot1agCfmLtrIngress	The value returned in the Ingress Action Field of the LTM.

Table/Group	Supported Variables	Comments
	dot1agCfmLtrIngressMac	MAC address returned in the ingress MAC address field.
	dot1agCfmLtrIngressPortIdSubtype	interfaceName (5) is returned if present.
	dot1agCfmLtrIngressPortId	interfaceName (5), then the octet string identifies a particular instance of the ifName object. If the particular ifName object does not contain any values, another port identifier type should be used.
	dot1agCfmLtrEgress	The value returned in the Egress Action Field of the LTM.
	dot1agCfmLtrEgressMac	MAC address returned in the egress MAC address field.
	dot1agCfmLtrEgressPortIdSubtype	interfaceName (5) is returned if present.
	dot1agCfmLtrEgressPortId	interfaceName (5), then the octet string identifies a particular instance of the ifName object. If the particular ifName object does not contain any values, another port identifier type should be used.
	dot1agCfmLtrOrganizationSpecificTlv	All organization specific TLVs returned in the LTR.
dot1agCfmStackTable	dot1agCfmStackifIndex	Index object. This object represents the bridge port or aggregated port on which MEPs or MHFs might be configured.
	dot1agCfmStackVlanIdOrNone	Index object. VLAN ID to which the MP is attached.
	dot1agCfmStackMdLevel	Index object. MD Level of the Maintenance Point.
	dot1agCfmStackDirection	Index object. Direction in which the MP faces on the bridge port.
	dot1agCfmStackMdIndex	The index of the MD in the dot1agCfmMdTable to which the MP is associated.

Table/Group	Supported Variables	Comments
	dot1agCfmStackMaIndex	The index of the MA in the dot1agCfmMaNetTable and dot1agCfmMaCompTable to which the MP is associated.
	dot1agCfmStackMepId	If an MEP is configured, the MEP ID.
	dot1agCfmStackMacAddress	MAC address of the MP.
dot1agCfmMaMepListTable	dot1agCfmMdIndex	MD Index.
	dot1agCfmMaIndex	MA Index.
	dot1agCfmMaMepListIdentifier	MEP identifier.
	dot1agCfmMaMepListRowStatus	

Table/Group	Supported Variables	Comments
dot1agCfmFaultAlarm (NOTIFICATION)	dot1agCfmMepHighestPrDefect	A MEP has a persistent defect condition. A notification (fault alarm) is sent to the management entity with the OID of the MEP that has detected the fault. The management entity receiving the notification can identify the system from the network source address of the notification, and can identify the MEP reporting the defect by the indices in the OID of the dot1agCfmMepHighestPrDefect variable in the notification: dot1agCfmMdIndex—Also the index of the MEPs.MD table entry (dot1agCfmMdTable).dot1agCfmMaIndex—Also an index (with the MD table index) of the MEP's MA network table entry (dot1agCfmMaNetTable), and (with the MD table index and component ID) of the MEP's MA component table entry.dot1agCfmMepIdentifier—MEP Identifier and final index into the MEP table (dot1agCfmMepTable).
extremeCfmMibConformance		
dot1agCfmMaCompTable		Not supported
dot1agCfmConfigErrorList Table		Not supported
dot1agCfmVlanTable		Not supported
dot1agCfmDefaultMdTable		Not supported

IEEE8021-PAE-MIB

This MIB contains objects for the 802.1X protocol draft D10 of the 802.1X standard.

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
dot1xPaeSystemAuthControl		
dot1xPaePortTable	All objects	
dot1xAuthConfigTable	Not supported	In lieu of these tables, Extreme Networks supports the per-station based versions which are present in the IEEE8021X- EXTENSIONS-MIB.
dot1xAuthStatsTable	Not supported	
dot1xAuthDiagTable	Not supported	This table has been deprecated in the drafts subsequent to the 2001 version of the 802.1X standard.
dot1xAuthSessionStatsTable	Not supported	
dot1xSuppConfigTable	None	These tables are not applicable to the switch since they are for a supplicant.
dot1xSuppStatsTable	None	

IEEE8021X-EXTENSIONS-MIB

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
dot1xAuthStationTable	All objects	
dot1xAuthConfigTable	All objects	
dot1xAuthStatsTable	All objects	

ISIS-MIB (draft-ietf-isis-wg-mib-10.txt)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
IsisSysTable	isisSysInstance	
	isisSysVersion	
	isisSysType	
	isisSysID	

Table/Group	Supported Variables	Comments
	isisSysMaxPathSplits	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysMaxLSPGenInt	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysMaxAreaAddresses	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysPollESHelloRate	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysWaitTime	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysAdminState	
	isisSysLogAdjacencyChanges	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysNextCircIndex	
	isisSysL2toL1Leaking	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysMaxAge	
	isisSysReceiveLSPBufferSize	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysExistState	Only state destroy (6) is supported.
isisSysLevelTable	isisSysLevelIndex	
	isisSysLevelOrigLSPBuffSize	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisSysLevelMinLSPGenInt	Supported isisSysLevelMinLSPGenInt range is 1 - 120 (MIB is 1 - 65535).
	isisSysLevelOverloadState	
	isisSysLevelSetOverload	
	isisSysLevelSetOverloadUntil	
	isisSysLevelMetricStyle	
	isisSysLevelSPFConsiders	
isisManAreaAddrTable	isisManAreaAddr	
	isisManAreaAddrExistState	

Table/Group	Supported Variables	Comments
isisAreaAddrTable	isisAreaAddr	isisAreaAddrTable only displays the contents of the isisManAreaAddrTable.
isisSysProtSuppTable	isisSysProtSuppProtocol	
	isisSysProtSuppExistState	Supported as read-only.
isisSummAddrTable	isisSummAddressType	
	isisSummAddress	
	isisSummAddrPrefixLen	
	isisSummAddrExistState	
	isisSummAddrMetric	Supported as read-only.
	isisSummAddrFullMetric	Supported as read-only.
isisCircTable	isisCircIndex	
	isisCircIfIndex	Supported as read-only.
	isisCircIfSubIndex	Supported as read-only.
	isisCircAdminState	Setting isisCircAdminState to off (2) deletes the entry.
	isisCircExistState	Only state destroy (6) is supported.
	isisCircType	Only broadcast (1) and ptToPt (2) are supported for isisCircType.
	isisCircExtDomain	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisCircLevel	
	isisCircPassiveCircuit	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisCircMeshGroupEnabled	
	isisCircMeshGroup	
	isisCircSmallHellos	
	isisCircLastUpTime	
	isisCirc3WayEnabled	Supported as read-only.
isisCircLevelTable	isisCircLevelIndex	
	isisCircLevelMetric	Supported isisCircLevelMetric range is 1 - 63 (MIB is 0 - 63).
	isisCircLevelWideMetric	Supported isisCircLevelMetricWide range is 1 - 16777214 (MIB is 0 - 16777214).
	isisCircLevelISPriority	

Table/Group	Supported Variables	Comments
	isisCircLevelIDOctet	Supported as read-only.
	isisCircLevelID	
	isisCircLevelDesIS	
	isisCircLevelHelloMultiplier	Default isisCircLevelHelloMultiplier is 3 (MIB is 10).
	isisCircLevelHelloTimer	Default isisCircLevelHelloTimer is 10000 ms (MIB is 3000 ms). Supported range is 1000 - 600000 ms (MIB is 10 - 600000 ms). Fractions of a second are rounded to the nearest whole second.
	isisCircLevelDRHelloTimer	Supported as read-only. isisCircLevelDRHelloTimer object is computed as 1/3 of isisCircLevelHelloTimer.
	isisCircLevelLSPThrottle	Default isisCircLevelLSPThrottle is 33 ms (MIB is 30 ms).
	isisCircLevelMinLSPRetransInt	
	isisCircLevelCSNPInterval	
	isisCircLevelPartSNPInterval	Unsupported object. Always returns MIB default value if one exists for a GET.
isisSystemCounterTable	isisSysStatLevel	
	isisSysStatCorrLSPs	
	isisSysStatAuthTypeFails	
	isisSysStatAuthFails	
	isisSysStatLSPDbaseOloads	
	isisSysStatManAddrDropFromAreas	
	isisSysStatAttmptToExMaxSeqNums	
	isisSysStatSeqNumSkips	
	isisSysStatOwnLSPPurges	
	isisSysStatIDFieldLenMismatches	
	isisSysStatMaxAreaAddrMismatches	
	isisSysStatPartChanges	
	isisSysSPFRuns	

Table/Group	Supported Variables	Comments
isisCircuitCounterTable	isisCircuitType	
	isisCircAdjChanges	
	isisCircNumAdj	
	isisCircInitFails	
	isisCircRejAdjs	
	isisCircIDFieldLenMismatches	
	isisCircMaxAreaAddrMismatches	
	isisCircAuthTypeFails	
	isisCircAuthFails	
	isisCircLANDesISChanges	
isisPacketCounterTable	isisPacketCountLevel	
	isisPacketCountDirection	
	isisPacketCountIIHello	
	isisPacketCountISHello	
	isisPacketCountESHello	
	isisPacketCountLSP	
	isisPacketCountCSNP	
	isisPacketCountPSNP	
	isisPacketCountUnknown	
isisISAdjTable	isisISAdjIndex	
	isisISAdjState	
	isisISAdj3WayState	
	isisISAdjNeighSNPAAddress	
	isisISAdjNeighSysType	
	isisISAdjNeighSysID	
	isisISAdjExtendedCircID	
	isisISAdjUsage	
	isisISAdjHoldTimer	
	isisISAdjNeighPriority	
	isisISAdjLastUpTime	
isisISAdjAreaAddrTable	isisISAdjAreaAddrIndex	
	isisISAdjAreaAddress	

Table/Group	Supported Variables	Comments
isisISAdjIPAddrTable	isisISAdjIPAddrIndex	
	isisISAdjIPAddressType	
	isisISAdjIPAddress	
isisISAdjProtSuppTable	isisISAdjProtSuppProtocol	
isisIPRATable	isisIPRAIndex	
	isisIPRADestType	
	isisIPRADest	
	isisIPRADestPrefixLen	
	isisIPRANextHopType	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisIPRANextHop	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisIPRAType	Supported as read-only.
	isisIPRAExistState	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisIPRAAdminState	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisIPRAMetric	Supported as read-only.
	isisIPRAMetricType	Supported as read-only.
	isisIPRAFullMetric	Supported as read-only.
	isisIPRASNPAAddress	Unsupported object. Always returns MIB default value if one exists for a GET.
	isisIPRASourceType	
isisLSPSummaryTable	isisLSPLevel	
	isisLSPId	
	isisLSPSeq	
	isisLSPZeroLife	
	isisLSPChecksum	
	isisLSPLifetimeRemain	
	isisLSPPDULength	
	isisLSPAttributes	
isisLSPTLVTable	isisLSPTLVIndex	

Table/Group	Supported Variables	Comments
	isisLSPTLVSeq	
	isisLSPTLVChecksum	
	isisLSPTTLVType	
	isisLSPTLVLen	
	isisLSPTLVValue	

OSPFv3-MIB (draft-ietf-ospf-ospfv3-mib-10)

All tables and variables of this MIB are supported.

Note
SNMP SET requests for *OSPFv3 (Open Shortest Path First version 3)* MIB objects are not supported.

PIM-MIB (draft-ietf-pim-mib-v2-01.txt)

This MIB is superset of RFC 2934.

Table/Group	Supported Variables	Comments
pimInterfaceTable	pimInterfaceIfIndex	
	pimInterfaceAddress	
	pimInterfaceNetMask	
	pimInterfaceMode	
	pimInterfaceDR	
	pimInterfaceHelloInterval	
	pimInterfaceStatus	
	pimInterfaceJoinPruneInterval	
	pimInterfaceCBSRPreference	
	pimInterfaceTrigHelloInterval	Not supported.
	pimInterfaceHelloHoldtime	These objects are supported as read only.
	pimInterfaceLanPruneDelay	
	pimInterfacePropagationDelay	
	pimInterfaceOverrideInterval	
	pimInterfaceGenerationID	
	pimInterfaceJoinPruneHoldtime	

Table/Group	Supported Variables	Comments
	pimInterfaceGraftRetryInterval	
	pimInterfaceMaxGraftRetries	
	pimInterfaceSRTTLThreshold	
	pimInterfaceLanDelayEnabled	
	pimInterfaceSRCapable	
	pimInterfaceDRPriority	This object is supported as read only.
pimNeighborTable	pimNeighborAddress	
	pimNeighborIfIndex	
	pimNeighborUpTime	
	pimNeighborExpiryTime	
	pimNeighborMode	Feature unsupported, only default value is returned.
	pimNeighborLanPruneDelay	Feature unsupported, only default value is returned.
	pimNeighborOverrideInterval	
	pimNeighborTBit	Feature unsupported so only default value is returned.
	pimNeighborSRCapable	Feature unsupported so only default value is returned.
	pimNeighborDRPresent	Feature unsupported so only default value is returned.
pimIpMRouteTable	pimIpMRouteUpstreamAssertTimer	
	pimIpMRouteAssertMetric	
	pimIpMRouteAssertMetricPref	
	pimIpMRouteAssertRPTBit	
	pimIpMRouteFlags	
	pimIpMRouteRPFNeighbor	
	pimIpMRouteSourceTimer	
	pimIpMRouteOriginatorSRTTL	Feature unsupported so only default value is returned.
pimIpMRouteNextHopTable	pimIpMRouteNextHopPruneReason	
	pimIpMRouteNextHopAssertWinner	

Table/Group	Supported Variables	Comments
	pimIpMRouteNextHopAssertTimer	
	pimIpMRouteNextHopAssertMetric	Not supported.
	pimIpMRouteNextHopAssertMetricPref	Not supported.
	pimIpMRouteNextHopJoinPruneTimer	Not supported.
pimRPSetTable	pimRPSetGroupAddress	
	pimRPSetGroupMask	
	pimRPSetAddress	
	pimRPSetHoldTime	
	pimRPSetExpiryTime	
	pimRPSetComponent	
pimCandidateRPTable	pimCandidateRPGroupAddress	
	pimCandidateRPGroupMask	
	pimCandidateRPAddress	
	pimCandidateRPRowStatus	
pimComponentTable	pimComponentIndex	
	pimComponentBSRAddress	
	pimComponentBSRExpiryTime	
	pimComponentCRPHoldTime	This object is supported as read only.
	pimComponentStatus	
Scalars	pimJoinPruneInterval	
	pimSourceLifetime	State Refresh feature is not supported, so these variables are set to defaults.
	pimStateRefreshInterval	
	pimStateRefreshLimitInterval	
	pimStateRefreshTimeToLive	
PIM Traps	pimNeighborLoss	Not supported.

SNMPv3 MIBs

The ExtremeXOS *SNMP* stack fully supports the SNMPv3 protocol and therefore implements the MIBs in the SNMPv3 RFCs.

Specifically, the MIBs in following RFCs are fully supported:

- RFC 2576—Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
- RFC 3410—Introduction and Applicability Statements for Internet-Standard Management Framework
- RFC 3411—An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks
- RFC 3412—Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
- RFC 3413—Simple Network Management Protocol (SNMP) Applications.
- RFC 3414—User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
- RFC 3415—View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
- RFC 3826—The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model

RFC 1213 (MIB-II)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
System group scalars	All objects	The sysServices object always returns the value 79. **Note:** The allowed character set is A-Z, a-z, 0-9, +-@_.,:;()/ ” for sysName, sysLocation, sysContact.
Interfaces group		Supported as per RFC 2233.
IP Group scalars	All objects	
ipAddrTable	All objects	
ipRouteTable	All objects	Supported as read only. Routes are indexed by prefix only.

Table/Group	Supported Variables	Comments
ipNetToMediaTable	All objects	context support is available for the ipNetToMediaTable in MIB-2. In order to retrieve entries in various VRs for ipNetToMediaTable in MIB-2, VRs can be mentioned as contexts. For example, to get the entries in *VR-Mgmt*, the following net-snmp query needs to be used: `snmpwalk -v3 -n "VR-Mgmt" -u <v3username> -a md5 -a <v3user-authentication password> <deviceIpAddress/Hostname> ipNetToMediaTable` Here -n has the context value of "VR-Mgmt". If no context is specified in the *SNMP* query, then only the entries in the *VR-Default* are shown.
ICMP group	All objects	
TCP group scalars	All objects	
tcpConnTable	All objects	
UDP group scalars	All objects	
udpTable	All objects	
EGP Group	Not supported	
SNMP group	All objects	
At group	All objects	Supported as read only.

History

Release version	History
NOS 7.2.0	This MIB was introduced.

RFC 1215

This MIB defines an SMI for SNMPv1 traps, and some traps themselves.

Of these, the following are supported.

Traps	Comments
coldStart	
warmStart	
authenticationFailure	The authentication failure trap will have additional extreme proprietary varbinds (extremeAuthFailSrcAddr, extremeAuthFailSrcAddressType , extremeAuthFailSrcAddress, extremeAuthFailSrcAddressVrName)

Traps	Comments
linkDown	
linkUp	

RFC 1493 (BRIDGE-MIB) and draft-ietf-bridge-rstpmib-03.txt

The BRIDGE-MIB has been augmented with draft-ietf-bridge-rstpmib-03.txt for 802.1w support.

Objects below that are defined in the latter are marked as such.

Table/Group	Supported Variables	Comments
dot1dBase group scalars	dot1dBaseBridgeAddress	
	dot1dBaseNumPorts	This object returns the number of ports in *STP* domain s0, not the total number of ports on the switch.
	dot1dBaseType	
dot1dBasePortTable	dot1dBasePort dot1dBasePortIfIndex dot1dBasePortCircuit dot1dBasePortMtuExceededDiscards	dot1BasePort is supported and returns the port number. dot1dBasePortIfIndex is supported. dot1dBasePortCircuit always has the value { 0 0 } since there is a one to one correspondence between a physical port and its ifIndex. dot1dBasePortMtuExceededDiscards is supported and value can be checked if packets exceeds MTU configured on particular ports.
dot1dStp group scalars	dot1dStpProtocolSpecification	Values for these objects will be returned for the STP domain s0 only. For other domains, see the EXTREME-STPEXTENSTIONS-MIB.
	dot1dStpPriority	
	dot1dStpTimeSinceTopologyChange	
	dot1dStpTopChanges	
	dot1dStpDesignatedRoot	
	dot1dStpRootCost	
	dot1dStpRootPort	
	dot1dStpMaxAge	
	dot1dStpHelloTime	
	dot1dStpHoldTime	
	dot1dStpForwardDelay	
	dot1dStpBridgeMaxAge	

Table/Group	Supported Variables	Comments
	dot1dStpBridgeHelloTime	
	dot1dStpBridgeForwardDelay	
	dot1dStpVersion	This object is not present in the original RFC1493, but is defined in the Internet draft draft-ietf-bridge-rstp-mib-03.txt.
	dot1dStpTxHoldCount	This object is not present in the original RFC1493, but is defined in the Internet draft draft-ietf-bridge-rstp-mib-03.txt. This object not supported; it always returns a value of (1). Attempting to set it yields an error.
	dot1dStpPathCostDefault	This object is not present in the original RFC1493, but is defined in the Internet draft draft-ietf-bridge-rstp-mib-03.txt. For this object only 8021d1998 (1) is supported at this time, not stp802112001 (2). Attempting to set (2) yields an error.
dot1dStpExtPortTable	All objects	This object is not present in the original RFC1493, but is defined in the Internet draft draft-ietf-bridge-rstp-mib-03.txt. The object 'dot1dStpPortProtocolMigration' is not supported; it always returns a value of (2). Attempting to set it yields an error.
dot1dStpPortTable	All objects	
STP Traps	newRoot	
	topologyChange	
dot1dTpFdbTable	Supported	The object dot1dTpFdbTable displays ports and *FDB* MAC addresses. They include both the static and dynamic FDB entries on the switch. The MIB does not provide a way to identify the *VLAN* on which the entry was learned. The port numbers are assumed to be 1 to 128 on Slot 1, and 128 to 255 on Slot 2, etc.
dot1dTpPortTable	Supported	
dot1dStatic group	Supported	
	Dot1dStaticAllowedToGoTo	Not supported

RFC 4363 (P-BRIDGE-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
dot1dPortCapabilitiesTable	dot1dDeviceCapabilities.	A table that contains capabilities information about every port that is associated with this bridge. Read-only.
dot1dDeviceCapabilities	Bit 0 (dot1dExtendedFilteringServices)	Denotes the multicast address filtering capability by GMRP. ExtremeXOS does not support this, and returns 0 always.
	Bit 1 (dot1dTrafficClasses)	Denotes switch capability on user priority mapping for multiple traffic classes. ExtremeXOS supports this, and always returns 1.
	Bit 2 (dot1qStaticEntryIndividualPort)	Denotes switch capability on Static unicast/multicast address support for all/particular port. ExtremeXOS does not support this, and returns 0 always.
	Bit 3 (dot1qIVLCapable)	Denotes switch capability on Independent VLAN Learning (IVL). ExtremeXOS supports this, and always returns 1.
	Bit 4 (dot1qSVLCapable)	Denotes switch capability on Shared VLAN Learning (SVL). ExtremeXOS does not support this, and returns 0 always.
	Bit 5 (dot1qHybridCapable)	Denotes switch capability to support both IVL and SVL. ExtremeXOS does not support this, and returns 0 always.
	Bit 6 (dot1qConfigurablePvidTagging)	Denotes whether the switch supports the ability to override the default PVID setting and its egress status (VLAN-Tagged or Untagged) on each port. ExtremeXOS supports this, and always returns 1.
	Bit 7 (dot1dLocalVlanCapable)	Denotes whether the switch supports multiple local vlans.ExtremeXOS supports this, and always returns 1.
dot1dPortDefaultUserPriority		The default ingress User Priority for this port. This only has effect on media, such as Ethernet, that do not support native User Priority. The value of this object MUST be retained across reinitializations of the management system. Read-write access.

RFC 4363 (Q-BRIDGE-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
RFC 4363 dot1qBasegroup Read-only access	dot1qVlanVersionNumber	The version number of IEEE 802.1Q that this device supports. Read-only access.
	dot1qMaxVlanId	The maximum IEEE 802.1Q *VLAN*-ID that this device supports. Read-only access.
	dot1qMaxSupportedVlans	The maximum number of IEEE 802.1Q VLANs that this device supports. Read-only access.
	dot1qNumVlans	The current number of IEEE 802.1Q VLANs that are configured in this device. Read-only access.
	dot1qGvrpStatus	The administrative status requested by management for GVRP. The value enabled (1) indicates that GVRP should be enabled on this device, on all ports for which it has not been specifically disabled. When disabled (2), GVRP is disabled on all ports, and all GVRP packets is forwarded transparently. This object affects all GVRP Applicant and Registrar state machines. A transition from disabled (2) to enabled (1) causes a reset of all GVRP state machines on all ports. The value of this object MUST be retained across reinitializations of the management system. Read-write access.

Table/Group	Supported Variables	Comments
dot1qPortVlanTable Read-only access	dot1qPvid	The PVID, the VLAN-ID assigned to untagged frames or Priority-Tagged frames received on this port. Read-Write access.
	dot1qPortAcceptableFrameTypes	When admitAll (1), untagged frames or Priority-Tagged frames received on this port will be accepted and assigned to a VID based on the PVID and VID Set for this port. This control does not affect VLAN-independent Bridge Protocol Data Unit (BPDU) frames, such as GVRP and Spanning Tree Protocol (*STP*). It does affect VLAN- dependent BPDU frames, such as GMRP. ExtremeXOS does not support on setting of this object with admitOnlyVlanTagged (2). Read-write access.
	dot1qPortIngressFiltering	When this is true (1), the device discards incoming frames for VLANs that do not include this Port in its Member set. When false (2), the port accepts all incoming frames. This control does not affect VLAN-independent BPDU frames, such as GVRP and STP. It does affect VLAN-dependent BPDU frames such as GMRP. The value of this object MUST be retained across reinitializations of the management system. **Note:** Note: Ingress filtering cannot be disabled on this object. Read-Write access.
	dot1qPortGvrpStatus	The state of GVRP operation on this port. The value enabled (1) indicates that GVRP is enabled on this port, as long as dot1qGvrpStatus is also enabled for this device. When disabled (2) but dot1qGvrpStatus is still enabled for the device, GVRP is disabled on this port: any GVRP packets received is silently discarded, and no GVRP registrations is propagated from other ports. This object affects all GVRP Applicant and Registrar state machines on this port. A transition from disabled (2) to enabled (1) causes a reset of all GVRP state machines on this port. The value of this object MUST be retained across reinitializations of the management system. Read-write access.
	dot1qPortGvrpFailedRegistrations	The total number of failed GVRP registrations, for any reason, on this port.

Table/Group	Supported Variables	Comments
		Read-only access
	dot1qPortGvrpLastPduOrigin	The Source MAC Address of the last GVRP message received on this port. Read-only access.
	dot1qPortRestrictedVlanRegistration	This object is always set to FALSE in ExtremeXOS. Restricted VLAN registration per port cannot be enabled. Read-write access.

Table/Group	Supported Variables	Comments
dot1qVlanStaticTable	dot1qVlanStaticName: Supported dot1qVlanStaticEgressPorts: Supported, but interpreted to mean that the specified ports are egress and ingress members of the VLAN. ExtremeXOS does not implement separate egress and ingress state. dot1qVlanForbiddenEgressPorts: Not supported, and SNMP GET always returns zeroes. dot1qVlanStaticUntaggedPorts: Supported, but interpreted to mean that the specified ports are egress and ingress untagged members of the VLAN. ExtremeXOS does not implement separate egress and ingress state. dot1qVlanStaticRowStatus: Only SNMP SET of createAndGo(4) and Destroy(6) are implemented. createAndWait(5) behaves like createAndGo(4).	An administratively assigned string, which may be used to identify the VLAN. Read-Create access. The set of ports that are permanently assigned to the egress list for this VLAN. Read-Create access. The set of ports that are prohibited by management from being included in the egress list for this VLAN. ExtremeXOS does not support forbidden ports, so this object always returns 0. Read-Create Access. The set of ports that should transmit egress packets for this VLAN as untagged. Read-Create access. This object indicates the status of this entry. Read-Create access.
dot1qVlanCurrentTable	dot1qVlanFdbId dot1qVlanCurrentEgressPorts dot1qVlanCurrentUntaggedPorts dot1qVlanStatus dot1qVlanCreationTime	A table containing current configuration information for each VLAN currently configured into the device by (local or network) management, or dynamically created as a result of GVRP requests received." The objects in this table has read-only access.

RFC 1724 (RIPv2-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
rip2Globals	rip2GlobalRouteChanges	
	rip2GlobalQueries	
rip2IfStatTable	All objects	
rip2IfConfTable	All objects	
rip2PeerTable	Not supported	

RFC 1850 (OSPF-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
ospfGeneralGroup	All objects	
ospfAreaTable	All objects	
ospfStubAreaTable	All objects	
ospfLsdbTable	All objects	
ospfAreaRangeTable	All objects	
ospfHostTable	All objects	
ospfIfTable	All objects	
ospfIfMetricTable	All objects	
ospfVirtIfTable	All objects	
ospfNbrTable	All objects	
ospfVirtNbrTable	All objects	
ospfExtLsdbTable	All objects	
ospfAreaAggregateTable	All objects	
ospfTrap	All traps	

RFC 2021 (RMON2-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
	probeCapabilities	
	probeSoftwareRev	
	probeHardwareRev	

Table/Group	Supported Variables	Comments
	probeDateTime	
	probeResetControl	
trapDestTable	All objects	

RFC 2233 (IF-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
	IfNumber	
ifTable	ifIndex	The ifIndex for ports is calculated as ((slot * 1000) + port). For VLANs, the ifIndex starts from 1000001.
	ifDescr	
	ifType	Only the following values are supported: {other, ethernetCsmacd, softwareLoopback, propVirtual}
	ifMtu	
	ifSpeed	
	ifPhysAddress	
	ifAdminStatus	The testing state is not supported.
	ifOperStatus	
	ifLastChange	
	ifInOctets	Updated every time queries this counter.
	ifInUcastPkts	Updated every time SNMP queries this counter.
	IfInNUcastPkts(deprecated)	Though deprecated, this object returns a value if the system keeps a count. Updated every time SNMP queries this counter.
	ifInDiscards	No count is kept of this object, so it always returns 0.
	ifInErrors	Updated every time SNMP queries this counter.
	ifInUnknownProtos	No count is kept of this object, so it always returns 0.
	ifOutOctets	Updated every time SNMP queries this counter.
	ifOutUcastPkts	Updated every time SNMP queries this counter.

Table/Group	Supported Variables	Comments
	IfOutNUcastPkts(deprecated)	Though deprecated, this object returns a value if the system keeps a count. Updated every time SNMP queries this counter.
	ifOutDiscards	No count is kept of this object, so it always returns 0.
	ifOutErrors	Updated every time SNMP queries this counter.
	IfOutQLen(deprecated)	Though deprecated, this object will returns a value if the system keeps a count. Updated every time SNMP queries this counter.
	IfSpecific	Not implemented. Always returns iso.org.dod.internet.
ifXTable	All objects	Only port interfaces return non-zero values for the counter objects in this table. The object ifPromiscuousMode is supported as read-only. ifCounterDiscontinuityTime is not implemented. All the statistics counters in the ifXTable are updated every time SNMP queries them.
ifStackTable	Not supported	
IfTestTable	Not supported	
ifRcvAddressTable	All objects	ifRcvAddressTable is supported as read-only. Also, only entries for physical ports appear in it.
snmpTraps	linkDown	
	linkUp	

RFC 2613 (SMON)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
smonVlanStatsControlTable	smonVlanStatsControlIndex	A unique arbitrary index for this smonVlanStatsControlEntry.
	smonVlanStatsControlDataSource	The source of data for this set of *VLAN* statistics. This object MAY NOT be modified if the associated smonVlanStatsControlStatus object is equal to active (1).
	smonVlanStatsControlCreateTime	The value of sysUpTime when this control entry was last activated. This object allows a management station to detect deletion and recreation cycles between polls.

Table/Group	Supported Variables	Comments
	smonVlanStatsControlOwner	Administratively assigned named of the owner of this entry. It usually defines the entity that created this entry and is therefore using the resources assigned to it, though there is no enforcement mechanism, nor assurance that rows created are ever used.
	smonVlanStatsControlStatus	The status of this row. An entry MAY NOT exist in the active state unless all objects in the entry have an appropriate value. If this object is not equal to active (1), all associated entries in the smonVlanIdStatsTable SHALL be deleted.
smonVlanIdStatsTable	smonVlanIdStatsId	The unique identifier of the VLAN monitored for this specific statistics collection. Tagged packets match the VID for the range between 1 and 4094
	smonVlanIdStatsTotalPkts	The total number of packets counted on this VLAN.
	smonVlanIdStatsTotalOverflowPkts	The number of times the associated smonVlanIdStatsTotalPkts counter has overflowed.
	smonVlanIdStatsTotalHCPkts	The total number of packets counted on this VLAN.
	smonVlanIdStatsTotalOctets	The total number of octets counted on this VLAN.
	smonVlanIdStatsTotalOverflowOctets	The number of times the associated smonVlanIdStatsTotalOctets counter has overflowed.
	smonVlanIdStatsTotalHCOctets	The total number of octets counted on this VLAN.
	smonVlanIdStatsNUcastOverflow Octets	The number of times the associated smonVlanIdStatsNUcastOctets counter has overflowed.
	smonVlanIdStatsCreateTime	The value of sysUpTime when this entry was last activated. This object allows a management station to detect deletion and recreation cycles between polls.

Table/Group	Supported Variables	Comments
dataSourceDapsTable	dataSourceCapsObject	Defines an object that can be a SMON data source or a source or a destination for a port copy operation.
	dataSourceCopyCaps	PortCopy function capabilities of the specified dataSource. Note that these are static capabilities, which SHOULD NOT be adjusted because of current resources or configuration.
	dataSourceCapsIfIndex	This object contains the ifIndex value of the ifEntry associated with this smonDataSource. The agent MUST create 'propVirtual' ifEntries for each dataSourceCapsEntry of type VLAN or entPhysicalEntry.
portCopyConfigTable	portCopySource	The ifIndex of the source which has all packets redirected to the destination as defined by portCopyDest.
	portCopyDest	Defines the ifIndex destination for the copy operation.
	portCopyDestDropEvents	The total number of events in which port copy packets were dropped by the switch at the destination port due to lack of resources. Note that this number is not necessarily the number of packets dropped; it is just the number of times this condition has been detected. A single dropped event counter is maintained for each portCopyDest. Thus all instances associated with a given portCopyDest will have the same portCopyDestDropEvents value. The value for this field is “0” (zero) due to hardware limitation.

Table/Group	Supported Variables	Comments
	portCopyDirection	This object affects the way traffic is copied from a switch source port, for the indicated port copy operation. If this object has the value copyRxOnly (1), then only traffic received on the indicated source port will be copied to the indicated destination port. If this object has the value copyTxOnly (2), then only traffic transmitted out the indicated source port will be copied to the indicated destination port. If this object has the value copyBoth (3), then all traffic received or transmitted on the indicated source port will be copied to the indicated destination port. The creation and deletion of instances of this object is controlled by the portCopyRowStatus object. Note that there is no guarantee that changes in the value of this object performed while the associated portCopyRowStatus object is equal to active will not cause traffic discontinuities in the packet.
	portCopyStatus	Defines the status of the port copy entry. In order to configure a source to destination portCopy relationship, both source and destination interfaces MUST be present as an ifEntry in the ifTable and their respective ifAdminStatus and ifOperStatus values must be equal to up (1). If the value of any of those two objects changes after the portCopyEntry is activated, portCopyStatus transitions to notReady (3). The capability of an interface to be source or destination of a port copy operation is described by the copySourcePort (0) and copyDestPort (1) bits in dataSourceCopyCaps. Those bits should be appropriately set by the agent, in order to allow for a portCopyEntry to be created.

Table/Group	Supported Variables	Comments
smonPrioStatsControlTable		Not supported due to hardware limitations.
smonPrioStatsTable		

RFC 2665 (EtherLike-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
Dot3StatsTable	dot3StatsIndex	
	dot3StatsAlignmentErrors	
	dot3StatsFCSErrors	
	dot3StatsSingleCollisionFrames	
	dot3StatsMultipleCollisionFrames	
	dot3StatsSQETestErrors	Not supported
	dot3StatsDeferredTransmissions	
	dot3StatsLateCollisions	
	dot3StatsExcessiveCollisions	
	dot3StatsInternalMacTransmitErrors	
	dot3StatsCarrierSenseErrors	Not supported
	dot3StatsFrameTooLongs	
	dot3StatsInternalMacReceiveErrors	
	dot3StatsSymbolErrors	Not supported
	dot3StatsEtherChipSet	
	dot3StatsDuplexStatus	
dot3CollTable	dot3CollCount	
	dot3CollFrequencies	
dot3ControlTable		Not supported
dot3PauseTable		Not supported

Other Unsupported Tables and Nodes in EtherLike MIB:

dot3ControlTable, dot3PauseTable, dot3Tests – all nodes under this, dot3Errors, etherConformance, etherGroups, etherCompliance, dot3Compliance. RFC 1657 (Definitions of Managed Objects for the Fourth Version of the BGP-4 using SMIv2).

All tables and variables of this MIB are supported with read-only access.

RFC 2668 (MAU-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
ifMauTable	All objects	
ifJackTable	All objects	
ifMauAutoNegTable	All objects	Setting auto-negotiation through *SNMP* is not supported.

The following new Extreme proprietary MAU types have been added to the ifMauType textual convention:

```
extremeMauType1000BaseWDMHD OBJECT IDENTIFIER

    ::= { extremeMauType 7 }

            "Gigabit WDM, half duplex"

extremeMauType1000BaseWDMFD OBJECT IDENTIFIER

    ::= { extremeMauType 8 }

            "Gigabit WDM, full duplex"

extremeMauType1000BaseLX70HD OBJECT IDENTIFIER

    ::= { extremeMauType 9 }

            "Gigabit LX70, half duplex"

extremeMauType1000BaseLX70FD OBJECT IDENTIFIER

    ::= { extremeMauType 10 }

            "Gigabit LX70, full duplex"

extremeMauType1000BaseZXHD OBJECT IDENTIFIER

    ::= { extremeMauType 11 }

            "Gigabit ZX, half duplex"

extremeMauType1000BaseZXFD OBJECT IDENTIFIER

    ::= { extremeMauType 12 }

            "Gigabit ZX, full duplex"
```

```
extremeMauType1000BaseLX100HD OBJECT IDENTIFIER

    ::= { extremeMauType 13 }

            "Gigabit LX100, half duplex"

extremeMauType1000BaseLX100FD OBJECT IDENTIFIER

    ::= { extremeMauType 14 }

            "Gigabit LX100, full duplex"

extremeMauType10GBaseCX4 OBJECT IDENTIFIER

    ::= { extremeMauType 15 }

            "10 Gigabit CX4"

extremeMauType10GBaseZR OBJECT IDENTIFIER

    ::= { extremeMauType 16 }

            "10 Gigabit ZR"

extremeMauType10GBaseDWDM OBJECT IDENTIFIER

    ::= { extremeMauType 17 }

            "10 Gigabit DWDM"

extremeMauType10GBaseCX OBJECT IDENTIFIER

    ::= { extremeMauType 18 }

            "10 Gigabit CX - SFP+ twin coax cable"

extremeMauType10GBaseT OBJECT IDENTIFIER

    ::= { extremeMauType 19 }

            "10 Gigabit BaseT"

extremeMauType40G OBJECT IDENTIFIER

    ::= { extremeMauType 20 }

            "40 Gigabit interface"
```

Corresponding MAU Type List Bits values have been added:

```
extreme_ifMauTypeListBits_b1000baseWDMHD    -- 64

extreme_ifMauTypeListBits_b1000baseWDMFD    -- 65

extreme_ifMauTypeListBits_b1000baseLX70HD    -- 66

extreme_ifMauTypeListBits_b1000baseLX70FD    -- 67

extreme_ifMauTypeListBits_b1000baseZXHD    -- 68

extreme_ifMauTypeListBits_b1000baseZXFD    -- 69
```

The following standards-based additions have been made as a 'Work in Progress', as per draft-ietf-hubmib-mau-mib-v3-02.txt:

A new enumeration 'fiberLC(14)' for the JackType textual convention.

New MAU types:

```
dot3MauType10GigBaseX OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "X PCS/PMA (per 802.3 section 48), unknown PMD."

    ::= { dot3MauType 31 }

dot3MauType10GigBaseLX4 OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "X fiber over WWDM optics (per 802.3 section 53)"

    ::= { dot3MauType 32 }

dot3MauType10GigBaseR OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "R PCS/PMA (per 802.3 section 49), unknown PMD."

    ::= { dot3MauType 33 }

dot3MauType10GigBaseER OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "R fiber over 1550 nm optics (per 802.3 section 52)"

    ::= { dot3MauType 34 }

dot3MauType10GigBaseLR OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "R fiber over 1310 nm optics (per 802.3 section 52)"

    ::= { dot3MauType 35 }

dot3MauType10GigBaseSR OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "R fiber over 850 nm optics (per 802.3 section 52)"

    ::= { dot3MauType 36 }

dot3MauType10GigBaseW OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "W PCS/PMA (per 802.3 section 49 and 50), unknown PMD."

    ::= { dot3MauType 37 }
```

```
dot3MauType10GigBaseEW OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "W fiber over 1550 nm optics (per 802.3 section 52)"

    ::= { dot3MauType 38 }

dot3MauType10GigBaseLW OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "W fiber over 1310 nm optics (per 802.3 section 52)"

    ::= { dot3MauType 39 }

dot3MauType10GigBaseSW OBJECT-IDENTITY

    STATUS      current

    DESCRIPTION "W fiber over 850 nm optics (per 802.3 section 52)"

    ::= { dot3MauType 40 }
```

Corresponding new Mau Type List bit values:

```
    b10GbaseX(31)                       — 10GBASE-X

    b10GbaseLX4(32)                       — 10GBASE-LX4

    b10GbaseR(33)                       — 10GBASE-R

    b10GbaseER(34                       — 10GBASE-ER

    b10GbaseLR(35)                       — 10GBASE-LR

    b10GbaseSR(36)                       — 10GBASE-SR

    b10GbaseW(37)                       — 10GBASE-W

    b10GbaseEW(38)                       — 10GBASE-EW

    b10GbaseLW(39)                       — 10GBASE-LW

    b10GbaseSW(40)                       — 10GBASE-SW
```

RFC 2787 (VRRP-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
vrrpOperations	vrrpNodeVersion	
	vrrpNotificationCntl	
vrrpStatistics	vrrpRouterChecksumErrors	

Table/Group	Supported Variables	Comments
	vrrpRouterVersionErrors	
	vrrpRouterVrIdErrors	
vrrpOperTable	All objects	Creation of a new row or modifying an existing row requires vrrpOperAdminState to be set to down; otherwise any kind of set will fail on this table. vrrpOperAuthType does not support ipAuthenticationHeader.
vrrpAssoIpAddrTable	All objects	
vrrpRouterStatsTable	All objects	
vrrpNotifications	vrrpTrapNewMaster	
	vrrpTrapAuthFailure	

RFC-3433 (Entity Sensor MIB)

The Entity Sensor MIB contains a single entitySensorValueGroup that contains objects for representing values and status of the SFP/QSFP.

Table/Group	Supported Variables	Comments
entitySensorValueGroup	entPhySensorType	An integer that defines the type of data units for the sensor. Read-only.
	enetPhySensorScale	An integer representing the data scaling factor for the sensor. Read-only.
	entPhySensorPrecision	An integer representing the number of decimal places of sensor precision. Read-only.
	entPhySensorValue	An integer representing the sensor value. Read-only.
	entPhyOperStatus	Operational status of the sensor.
	entPhySensorUnitsDisplay	Text description of data units used in entPhySensorValue. Read-only.
	entPhySensorValueTimeStamp	Value of sysUpTime at the time the status and value was last obtained. Read-only.
	entPhySensorValueUpdateRate	Frequency that the agent updates the associated entPhySensorValue. Read-only.

entPhySensorType

The entPhySensorType object is an integer that defines the sensor as one of the following types:.Value

Value	Sensor Type
1	Other
2	Unknown
3	voltsAC
4	voltsDC
5	Amperes
6	Watts
7	Hertz
8	Celsius
9	percentRH
10	Rpm
11	Cmm
12	thruthvalue

The following sensor types are used for the digital diagnostic monitoring data.

Sensor	Sensor Type
Temperature	Celsius
VCC	voltsDC
Tx Bias Current	Amperes
Rx Power	Watts
Tx Power	Watts

Power measurements may be represented as decibels (dB). If this is the case, the sensor type is defined as “other” and the units of measurement are determined using entPhySensorUnitsDisplay.

entPhySensorScale

The entPhySensorScale object is an integer that defines the scale of the sensor as one of the following.

Value	Sensor Scale
1	Yocto
2	Zepto
3	Atto
4	femto

Value	Sensor Scale
5	Pico
6	nano
7	micro
8	milli
9	units
10	Kilo
11	mega
12	giga
13	tera
14	Exa
15	peta
16	zetta
17	yotta

entPhySensorPrecision

This object defines the sensor precision range.

A value in the range 1 to 9, represents the number of decimal places in the fractional part of an associated entPhySensorValue fixed-point number. A value in the range -8 to -1, represents the number of accurate digits in the associated entPhySensorValue fixed-point number. The value zero indicates the associated entPhySensorValue object is not a fixed-point number.

entPhySensorValue

The entPhySensorValue object is the value of the sensor.

entPhySensorOperStatus

The entPhySensorOperStatus object is an integer that defines the operating status of the sensor as one of the following.

Value	Sensor Status
1	Ok
2	Unavailable
3	Nonoperational

Ports without a populated transceiver or one that is unreachable due to i2c bus errors, etc. are designated “nonoperational”. Ports populated with a transceiver that does not support digital diagnostic monitoring are designated “unavailable.”

entPhySensorUnitsDisplay

The entPhySensorUnitsDisplay object is a textual description of the data units that should be used in the display of entPhySensorValue.

entPhySensorValueTimeStamp

The entPhySensorValueTimeStamp object is the value of sysUpTime at the time the status and/or value of this sensor were last obtained.

entPhySensorValueUpdateRate

The entPhySensorValueUpdateRate object is an indication of the frequency that the agent updates the associated entPhySensorValue object, represented in milliseconds.

RFC 3621 (PoE-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
pethPsePortTable	All objects	Objects in this table are read-only.
pethMainPseTable	All objects	Objects in this table are read-only.
pethNotifications	pethPsePortOnOffNotification	
	pethMainPowerUsageOnNotification	
	pethMainPowerUsageOffNotification	

RFC 4022 (TCP-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
tcpConnectionTable	tcpConnectionState	The state of this TCP connection.
	tcpConnectionProcess	The system's process ID for the process associated with this connection, or zero if there is no such process. This value is expected to be the same as HOST-RESOURCES-MIB::hrSWRunIndex or SYSAPPL-MIB::sysApplElmtRunIndex for some row in the appropriate tables.

Table/Group	Supported Variables	Comments
tcpListenerTable	All objects	A table containing information about TCP listeners. A listening application can be represented in three possible ways: 1. An application that is willing to accept both IPv4 and IPv6 datagrams is represented by a tcpListenerLocalAddressType of unknown (0) and a tcpListenerLocalAddress of ''h (a zero-length octet-string). 2. An application that is willing to accept only IPv4 or IPv6 datagrams is represented by a tcpListenerLocalAddressType of the appropriate address type and a tcpListenerLocalAddress of 0.0.0.0 or ::, respectively. 3. An application that is listening for data destined only to a specific IP address, but from any remote system, is represented by a tcpListenerLocalAddressType of an appropriate address type, with tcpListenerLocalAddress as the specific local address. **Note:** The address type in this table represents the address type used for the communication, irrespective of the higher-layer abstraction. For example, an application using IPv6 'sockets' to communicate via IPv4 between ::ffff:10.0.0.1 and ::ffff:10.0.0.2 would use InetAddressType ipv4(1)).
tcpBaseGroup	All objects	The group of counters common to TCP entities.

Table/Group	Supported Variables	Comments
tcpHCGroup	tcpHCInSegs	The total number of segments received, including those received in error. This count includes segments received on currently established connections. This object is the 64-bit equivalent of tcpInSegs. Discontinuities in the value of this counter are indicated via discontinuities in the value of sysUpTime.
	tcpHCOutSegs	The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets. This object is the 64-bit equivalent of tcpOutSegs. Discontinuities in the value of this counter are indicated via discontinuities in the value of sysUpTime.

RFC 4087 (TUNNEL-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
tunnelIfTable	tunnelIfAddressType	The type of address in the corresponding tunnelIfLocalInetAddress and tunnelIfRemoteInetAddress objects.
	tunnelIfEncapsLimit	The maximum number of additional encapsulations permitted for packets undergoing encapsulation at this node. A value of -1 indicates that no limit is present (except as a result of the packet size).
	tunnelIfLocalInetAddress	The address of the local endpoint of the tunnel (i.e., the source address used in the outer IP header). If the address is unknown, the value is 0.0.0.0 for IPv4 or :: for IPv6. The type of this object is given by tunnelIfAddressType.

Table/Group	Supported Variables	Comments
	tunnelIfRemoteInetAddress	The address of the remote endpoint of the tunnel (i.e., the destination address used in the outer IP header). If the address is unknown or the tunnel is not a point-to-point link (e.g., if it is a 6to4 tunnel), the value is 0.0.0.0 for tunnels over IPv4 or :: for tunnels over IPv6. The type of this object is given by tunnelIfAddressType.
tunnelInetConfigTable	tunnelInetConfigAddressType	The address type over which the tunnel encapsulates packets.
	tunnelInetConfigEncapsMethod	The encapsulation method used by the tunnel.
	tunnelInetConfigEntry	An entry (conceptual row) containing the information on a particular configured tunnel. Note that there is a 128 subid maximum for object OIDs. Implementers need to be aware that if the total number of octets in tunnelInetConfigLocalAddress and tunnelInetConfigRemoteAddress exceeds 110 then OIDs of column instances in this table will have more than 128 sub-identifiers and cannot be accessed using SNMPv2c or SNMPv3. In practice this is not expected to be a problem since IPv4 and IPv6 addresses will not cause the limit to be reached, but if other types are supported by an agent, care must be taken to ensure that the sum of the lengths do not cause the limit to be exceeded.

Table/Group	Supported Variables	Comments
	tunnelInetConfigID	An identifier used to distinguish between multiple tunnels of the same encapsulation method, with the same endpoints. If the encapsulation protocol only allows one tunnel per set of endpoint addresses (such as for GRE or IP-in-IP), the value of this object is: For encapsulation methods (such as L2F), which allow multiple parallel tunnels, the manager is responsible for choosing any ID which does not conflict with an existing row, such as choosing a random number.
	tunnelInetConfigIfIndex	If the value of tunnelInetConfigStatus for this row is active, then this object contains the value of ifIndex corresponding to the tunnel interface. A value of 0 is not legal in the active state, and means that the interface index has not yet been assigned.
	tunnelInetConfigLocalAddress	The address of the local endpoint of the tunnel, or 0.0.0.0 (for IPv4) or :: (for IPv6) if the device is free to choose any of its addresses at tunnel establishment time.
	tunnelInetConfigRemoteAddress	The address of the remote endpoint of the tunnel.

Table/Group	Supported Variables	Comments
	tunnelInetConfigStatus	Read-create. The status of this row, by which new entries may be created, or old entries deleted from this table. The agent need not support setting this object to createAndWait or notInService since there are no other writable objects in this table, and writable objects in rows of corresponding tables such as the tunnelIfTable may be modified while this row is active. To create a row in this table for an encapsulation method which does not support multiple parallel tunnels with the same endpoints, the management station should simply use a tunnelInetConfigID of 1, and set tunnelInetConfigStatus to createAndGo. For encapsulation methods such as L2F which allow multiple parallel tunnels, the management station may select a pseudo-random number to use as the tunnelInetConfigID and set tunnelInetConfigStatus to createAndGo. In event that this ID is already in use and an inconsistentValue is returned in response to the set operation, the management station should simply select a new pseudo-random number and retry the operation. Creating a row in this table will cause an interface index to be assigned by the agent in an implementation-dependent manner, and corresponding rows will be instantiated in the ifTable and the tunnelIfTable. The status of this row will become active as soon as the agent assigns the interface index, regardless of whether the interface is operationally up. Deleting a row in this table will likewise delete the corresponding row in the ifTable and in the tunnelIfTable.

Table/Group	Supported Variables	Comments
	tunnelInetConfigStorageType	Read-create. The storage type of this row. If the row is permanent (4), no objects in the row need be writable.
tunnelMIBInetGroup	All objects	A collection of objects to support basic management of IPv4 and IPv6 Tunnels.

RFC 4113 (UDP-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
udpBaseGroup	All objects	The group of objects providing for counters of UDP statistics.
udpEndpointGroup	All objects	The group of objects providing for the IP version independent management of UDP 'endpoints'.
udpEndpointTable	udpEndpointEntry	Information about a particular current UDP endpoint. Implementers need to be aware that if the total number of elements (octets or sub-identifiers) in udpEndpointLocalAddress and udpEndpointRemoteAddress exceeds 111, then OIDs of column instances in this table will have more than 128 sub-identifiers and cannot be accessed using SNMPv2c or SNMPv3.
	udpEndpointInstance	The instance of this tuple. This object is used to distinguish among multiple processes 'connected' to the same UDP endpoint. For example, on a system implementing the BSD sockets interface, this would be used to support the SO_REUSEADDR and SO_REUSEPORT socket options.

Table/Group	Supported Variables	Comments
	udpEndpointLocalAddress	The local IP address for this UDP endpoint. The value of this object can be represented in three possible ways, depending on the characteristics of the listening application: 1. For an application that is willing to accept both IPv4 and IPv6 datagrams, the value of this object must be ''h (a zero-length octet-string), with the value of the corresponding instance of the udpEndpointLocalAddressType object being unknown(0). 2. For an application that is willing to accept only IPv4 or only IPv6 datagrams, the value of this object must be 0.0.0.0 or ::, respectively, while the corresponding instance of the udpEndpointLocalAddressType object represents the appropriate address type. 3. For an application that is listening for data destined only to a specific IP address, the value of this object is the specific IP address for which this node is receiving packets, with the corresponding instance of the udpEndpointLocalAddressType object representing the appropriate address type. As this object is used in the index for the udpEndpointTable, implementors of this table should be careful not to create entries that would result in OIDs with more than 128 subidentifiers; else the information cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3.
	udpEndpointLocalAddressType	The address type of udpEndpointLocalAddress. Only IPv4, IPv4z, IPv6, and IPv6z addresses are expected, or unknown(0) if datagrams for all local IP addresses are accepted.
	udpEndpointLocalPort	The local port number for this UDP endpoint.
	udpEndpointProcess	The system's process ID for the process associated with this endpoint, or zero if there is no such process. This value is expected to be the same as HOST-RESOURCES-MIB::hrSWRunIndex or SYSAPPL-MIB:: sysApplElmtRunIndex for some row in the appropriate tables.

Table/Group	Supported Variables	Comments
	udpEndpointRemoteAddress	The remote IP address for this UDP endpoint. If datagrams from any remote system are to be accepted, this value is ''h (a zero-length octet-string). Otherwise, it has the type described by udpEndpointRemoteAddressType and is the address of the remote system from which datagrams are to be accepted (or to which all datagrams will be sent). As this object is used in the index for the udpEndpointTable, implementors of this table should be careful not to create entries that would result in OIDs with more than 128 subidentifiers; else the information cannot be accessed using SNMPv2c or SNMPv3.
	udpEndpointRemoteAddressType	The address type of udpEndpointRemoteAddress. Only IPv4, IPv4z, IPv6, and IPv6z addresses are expected, or unknown(0) if datagrams for all remote IP addresses are accepted. Also, note that some combinations of udpEndpointLocalAdressType and udpEndpointRemoteAddressType are not supported. In particular, if the value of this object is not unknown(0), it is expected to always refer to the same IP version as udpEndpointLocalAddressType.
	udpEndpointRemotePort	The remote port number for this UDP endpoint. If datagrams from any remote system are to be accepted, this value is zero.

Table/Group	Supported Variables	Comments
udpHCGroup	All objects	The group of objects providing for counters of high speed UDP implementations.
udpMIBCompliance2		The compliance statement for systems that implement UDP. There are a number of INDEX objects that cannot be represented in the form of OBJECT clauses in SMIv2, but for which we have the following compliance requirements, expressed in OBJECT clause form in this description clause: `-- OBJECT udpEndpointLocalAddressType -- SYNTAX InetAddressType { unknown(0), ipv4(1), -- ipv6(2), ipv4z(3), -- ipv6z(4) } -- DESCRIPTION -- Support for dns(5) is not required. -- OBJECT udpEndpointLocalAddress -- SYNTAX InetAddress (SIZE(0|4|8|16|20)) -- DESCRIPTION -- Support is only required for zero-length -- octet-strings, and for scoped and unscoped -- IPv4 and IPv6 addresses. -- OBJECT udpEndpointRemoteAddressType -- SYNTAX InetAddressType { unknown(0), ipv4(1), -- ipv6(2), ipv4z(3), -- ipv6z(4) } -- DESCRIPTION -- Support for dns(5) is not required. -- OBJECT udpEndpointRemoteAddress -- SYNTAX InetAddress (SIZE(0|4|8|16|20)) -- DESCRIPTION -- Support is only required for zero-length -- octet-strings, and for scoped and unscoped -- IPv4 and IPv6 addresses.`

RFC-4133 (ENTITY MIB)

For RFC-4133, the entityPhysicalGroup contains a single table, entityPhysicalTable, to identify physical system components. The following table lists the objects that are supported.

Table/Group	Description	Comments
entPhysicalIndex	The arbitrarily assigned index for the table	Not-accessible
entPhysicalDescr	Text description of the sensor.	Read-only
entPhysicalVendorType	Specified OID used to identify enterprise-wide type/version/model.	Read-only
entPhysicalContainedIn	The entPhysicalIndex of the parent device. For SFP sensors, this is the corresponding port index.	Read-only
entPhysicalClass	Integer that defines the general device type. SFP sensors are of class “sensor (8)”.	Read-only
entPhysicalParentRelPos	Relative position of the parent device.	Read-only
entPhysicaModulelName	Text String of the model information stored in the SFP/QSFP.	Read-only
entPhysicalSerialNum	Text String of the serial number information stored in the SFP/QSFP.	Read-only
entPhysicalMfgName	Manufacture Name of the SFP/QSFP, “Extreme Networks”.	Read-only

RFC 4292 (IP-FORWARD-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
inetCidrRouteTable	All objects	
inetCidrRouteDiscards		The number of valid route entries discarded from the inetCidrRouteTable. Discarded route entries do not appear in the inetCidrRouteTable. One possible reason for discarding an entry would be to free up buffer space for other route table entries.
inetCidrRouteNumber		The number of current inetCidrRouteTable entries that are not invalid.

RFC 4293 (IP-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
ipAddressTable	All objects	This table contains addressing information relevant to the entity's interfaces. This table does not contain multicast address information. Tables for such information should be contained in multicast specific MIBs, such as RFC 3019. While this table is writable, the user will note that several objects, such as ipAddressOrigin, are not. The intention in allowing a user to write to this table is to allow them to add or remove any entry that isn't permanent. The user should be allowed to modify objects and entries when that would not cause inconsistencies within the table. Allowing write access to objects, such as ipAddressOrigin, could allow a user to insert an entry and then label it incorrectly. **Note:** When including IPv6 link-local addresses in this table, the entry must use an InetAddressType of 'ipv6z' in order to differentiate between the possible interfaces.
ipSystemStatsTable	All objects	The number of valid route entries discarded from the inetCidrRouteTable. Discarded route entries do not appear in the inetCidrRouteTable. One possible reason for discarding an entry would be to free-up buffer space for other route table entries.

Table/Group	Supported Variables	Comments
ipAddressPrefixTable	All objects	This table allows the user to determine the source of an IP address or set of IP addresses, and allows other tables to share the information via pointer rather than by copying. For example, when the node configures both a unicast and anycast address for a prefix, the ipAddressPrefix objects for those addresses will point to a single row in this table. This table primarily provides support for IPv6 prefixes, and several of the objects are less meaningful for IPv4. The table continues to allow IPv4 addresses to allow future flexibility. In order to promote a common configuration, this document includes suggestions for default values for IPv4 prefixes. Each of these values may be overridden if an object is meaningful to the node. All prefixes used by this entity should be included in this table independent of how the entity learned the prefix. (This table isn't limited to prefixes learned from router advertisements.
ipv6ScopeZoneIndexTable	All objects	The table used to describe IPv6 unicast and multicast scope zones. For those objects that have names rather than numbers, the names were chosen to coincide with the names used in the IPv6 address architecture document.
ipDefaultRouterTable	All objects	The table used to describe the default routers known to this
ipv6RouterAdvertTable	All objects	The table containing information used to construct router advertisements.

RFC 5601 (PW-STD-MIB)

The following tables, groups, and variables are supported in this MIB.

All tables and variables of this MIB are supported as read only. The comments here are abbreviated versions of the description in the RFC documentation.

Table/Group	Supported Variables	Comments
pwTable	pwIndex	A unique index for the conceptual row identifying a PW within this table.
	pwPeerAddr	This object contains the value of the peer node address of the PW/PE maintenance protocol entity. This object should contain a value of all zeroes if not applicable (pwPeerAddrType is 'unknown').
	pwID	Pseudowire identifier. If the pwOwner object is pwIdFecSignaling or l2tpControlProtocol, then this object is signaled in the outgoing PW ID field within the Virtual Circuit FEC Element. For other values of pwOwner, this object is not signaled and it can be set to zero.
	pwLocalCapabAdvert	If a maintenance protocol is used, it indicates the capabilities the local node will advertise to the peer.
	pwRemoteGroupID	This object is obtained from the Group ID field as received through the maintenance protocol used for PW setup. Value of zero is reported if not used. Value of 0xFFFFFFFF is used if the object is yet to be defined by the PW maintenance protocol.
	pwOutboundLabel	The PW label used in the outbound direction (i.e., toward the PSN). It might be set manually if pwOwner is 'manual'; otherwise, it is set automatically.
	pwInboundLabel	The PW label used in the inbound direction (i.e., packets received from the PSN). It may be set manually if pwOwner is 'manual'; otherwise, it is set automatically.
	pwCreateTime	The value of sysUpTime at the time this PW was created.
	pwUpTime	Specifies the time since last change of pwOperStatus to Up (1).
	pwLastChange	The value of sysUpTime at the time the PW entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.
	pwAdminStatus	The desired operational status of this PW. This object can be set at any time.

Table/Group	Supported Variables	Comments
	pwOperStatus	This object indicates the operational status of the PW; it does not reflect the status of the Customer Edge (CE) bound interface.
	pwLocalStatus	Indicates the status of the PW in the local node.
	pwRemoteStatus	Indicates the status of the PW as was advertised by the remote.
	pwRowStatus	For creating, modifying, and deleting this row. This object can be changed at any time.
	pwOamEnable	This variable indicates if OAM is enabled for this PW. It can be changed at any time.
pwIndexMappingTable	All objects	This table enables the reverse mapping of the unique PWid parameters [peer IP, PW type, and PW ID] and the pwIndex. The table is not applicable for PWs created manually or by using the generalized FEC.

RFC 5602 (PW-MPLS-STD-MIB)

The following tables, groups, and variables are supported in this MIB.

All tables and variables of this MIB are supported as read only. The comments here are abbreviated versions of the description in the RFC documentation.

Table/Group	Supported Variables	Comments
pwMplsTable	pwIndex	This table controls MPLS-specific parameters when the PW is going to be carried over MPLS PSN.
	pwMplsMplsType	This object is set by the operator to indicate the outer tunnel types, if existing.
	pwMplsTtl	This object is set by the operator to indicate the PW TTL value to be used on the PW shim label.
	pwMplsLocalLdpID	The LDP identifier of the LDP entity that creates this PW in the local node.
	pwMplsLocalLdpEntityIndex	The local node LDP Entity Index of the LDP entity creating this PW.

Table/Group	Supported Variables	Comments
	pwMplsPeerLdpID	The peer LDP identifier of the LDP session. This object should return the value zero if LDP is not used or if the value is not yet known.
pwMplsOutboundTable	All objects	This table reports and configures the current outbound MPLS tunnels (i.e., toward the PSN) or the physical interface in the case of a PW label only that carries the PW traffic. It also reports the current outer tunnel and LSP that forward the PW traffic.

RFC 5603 (PW-ENET-STD-MIB)

The following tables, groups, and variables are supported in this MIB.

All tables and variables of this MIB are supported as read only. The comments here are abbreviated versions of the description in the RFC documentation.

Table/Group	Supported Variables	Comments
pwEnetTable	pwIndex	This table contains the index to the Ethernet tables associated with this Ethernet PW, the *VLAN* configuration, and the VLAN mode.
	pwEnetPwInstance	If multiple rows are mapped to the same PW, this index is used to uniquely identify the individual row.
	pwEnetPwVlan	This object defines the (service-delimiting) VLAN field value on the PW.
	pwEnetVlanMode	This object indicates the mode of VLAN handling between the port or the virtual port associated with the PW and the PW encapsulation.
	pwEnetPortVlan	This object defines if the mapping between the original port (physical port or VPLS virtual port) to the PW is VLAN based or not.
	pwEnetPortIfIndex	This object is used to specify the ifIndex of the Ethernet port associated with this PW for point-to-point Ethernet service, or the ifIndex of the virtual interface of the VPLS instance associated with the PW if the service is VPLS.
	pwEnetRowStatus	This object enables creating, deleting, and modifying this row.

RFC 5643

A subset of the traps in RFC 5643 are supported.

RFC 6933 (ENTITY-MIB)

The following tables, groups, and variables are supported in this MIB.

Table/Group	Supported Variables	Comments
entityPhysicalTable	entPhysicalIndex	The index for this entry.
	entPhysicalDescr	A textual description of physical entity. This object should contain a string that identifies the manufacturer's name for the physical entity and should be set to a distinct value for each version or model of the physical entity.
	entPhysicalVendorType	An indication of the vendor-specific hardware type of the physical entity. Note that this is different from the definition of MIB-II's sysObjectID. An agent should set this object to an enterprise-specific registration identifier value indicating the specific equipment type in detail. The associated instance of entPhysicalClass is used to indicate the general type of hardware device. If no vendor-specific registration identifier exists for this physical entity, or the value is unknown by this agent, then the value { 0 0 } is returned.
	entPhysicalContainedIn	The value of entPhysicalIndex for the physical entity that 'contains' this physical entity. A value of zero indicates this physical entity is not contained in any other physical entity. Note that the set of 'containment' relationships define a strict hierarchy; that is, recursion is not allowed. In the event that a physical entity is contained by more than one physical entity (e.g., double-wide modules), this object should identify the containing entity with the lowest value of entPhysicalIndex.

Table/Group	Supported Variables	Comments
	entPhysicalClass	An indication of the general hardware type of the physical entity. An agent should set this object to the standard enumeration value that most accurately indicates the general class of the physical entity, or the primary class if there is more than one entity. If no appropriate standard registration identifier exists for this physical entity, then the value 'other(1)' is returned. If the value is unknown by this agent, then the value 'unknown(2)' is returned.
	entPhysicalParentRelPos	An indication of the relative position of this 'child' component among all its 'sibling' components. Sibling components are defined as entPhysicalEntries that share the same instance values of each of the entPhysicalContainedIn and entPhysicalClass objects.
	entPhysicalName	The textual name of the physical entity. The value of this object should be the name of the component as assigned by the local device and should be suitable for use in commands entered at the device's 'console'. This might be a text name (e.g., 'console') or a simple component number (e.g., port or module number, such as '1'), depending on the physical component naming syntax of the device. If there is no local name, or if this object is otherwise not applicable, then this object contains a zero-length string. Note that the value of entPhysicalName for two physical entities will be the same in the event that the console interface does not distinguish between them, e.g., slot-1 and the card in slot-1.

Table/Group	Supported Variables	Comments
	entPhysicalHardwareRev	The vendor-specific hardware revision string for the physical entity. The preferred value is the hardware revision identifier actually printed on the component itself (if present). Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format in an implementation-specific manner. If no specific hardware revision string is associated with the physical component, or if this information is unknown to the agent, then this object will contain a zero-length string.
	entPhysicalFirmwareRev	The vendor-specific firmware revision string for the physical entity. Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format in an implementation-specific manner. If no specific firmware programs are associated with the physical component, or if this information is unknown to the agent, then this object will contain a zero-length string.
	entPhysicalSoftwareRev	The vendor-specific software revision string for the physical entity. Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format in an implementation-specific manner. If no specific software programs are associated with the physical component, or if this information is unknown to the agent, then this object will contain a zero-length string.

Table/Group	Supported Variables	Comments
	entPhysicalSerialNum	The vendor-specific serial number string for the physical entity. The preferred value is the serial number string actually printed on the component itself (if present). On the first instantiation of a physical entity, the value of entPhysicalSerialNum associated with that entity is set to the correct vendor-assigned serial number, if this information is available to the agent. If a serial number is unknown or non-existent, the entPhysicalSerialNum will be set to a zero-length string instead. Note that implementations that can correctly identify the serial numbers of all installed physical entities do not need to provide write access to the entPhysicalSerialNum object. Agents that cannot provide non-volatile storage for the entPhysicalSerialNum strings are not required to implement write access for this object. Not every physical component will have a serial number, or even need one. Physical entities for which the associated value of the entPhysicalIsFRU object is equal to 'false(2)' (e.g., the repeater ports within a repeater module) do not need their own unique serial numbers. An agent does not have to provide write access for such entities and may return a zero-length string. If write access is implemented for an instance of entPhysicalSerialNum and a value is written into the instance, the agent must retain the supplied value in the entPhysicalSerialNum instance (associated with the same physical entity) for as long as that entity remains instantiated. This includes instantiations across all re-initializations/reboots of the network management system, including those resulting in a change of the physical entity's entPhysicalIndex value.

Table/Group	Supported Variables	Comments
	entPhysicalMfgName	The name of the manufacturer of this physical component. The preferred value is the manufacturer name string actually printed on the component itself (if present). Note that comparisons between instances of the entPhysicalModelName, entPhysicalFirmwareRev, entPhysicalSoftwareRev, and the entPhysicalSerialNum objects are only meaningful amongst entPhysicalEntries with the same value of entPhysicalMfgName. If the manufacturer name string associated with the physical component is unknown to the agent, then this object will contain a zero-length string.
	entPhysicalModelName	The vendor-specific model name identifier string associated with this physical component. The preferred value is the customer-visible part number, which may be printed on the component itself. If the model name string associated with the physical component is unknown to the agent, then this object will contain a zero-length string.
	entPhysicalAlias	Not supported.
	entPhysicalAssetID	Not supported.
	entPhysicalIsFRU	This object indicates whether or not this physical entity is considered a 'field replaceable unit' by the vendor. If this object contains the value 'true(1)', then this entPhysicalEntry identifies a field replaceable unit. For all entPhysicalEntries that represent components permanently contained within a field replaceable unit, the value 'false(2)' should be returned for this object.
	entPhysicalMfgDate	This object contains the date of manufacturing of the managed entity. If the manufacturing date is unknown or not supported, the object is not instantiated. The special value '0000000000000000'H may also be returned in this case.

Table/Group	Supported Variables	Comments
	entPhysicalUris	This object contains identification information about the physical entity. The object contains URIs; therefore, the syntax of this object must conform to RFC 3986. Multiple URIs may be present and are separated by white space characters. Leading and trailing white space characters are ignored.If no URI identification information is known about the physical entity, the object is not instantiated. A zero-length octet string may also be returned in this case. **Note:** This object has read-write permission, but is implemented with read-only mode. Write operation is not supported.
	entPhysicalUUID	This object contains identification information about the physical entity. The object contains a Universally Unique Identifier, the syntax of this object must conform to RFC 4122, Section 4.1. A zero-length octet string is returned if no UUID information is known.
	entAliasMappingIdentifier	The value of this object identifies a particular conceptual row associated with the indicated entPhysicalIndex and entLogicalIndex pair.Because only physical ports are modeled in this table, only entries that represent interfaces or ports are allowed. If an ifEntry exists on behalf of a particular physical port, then this object should identify the associated ifEntry. For repeater ports, the appropriate row in the 'rptrPortGroupTable' should be identified instead.
	entPhysicalChildIndex	The value of entPhysicalIndex for the contained physical entity.

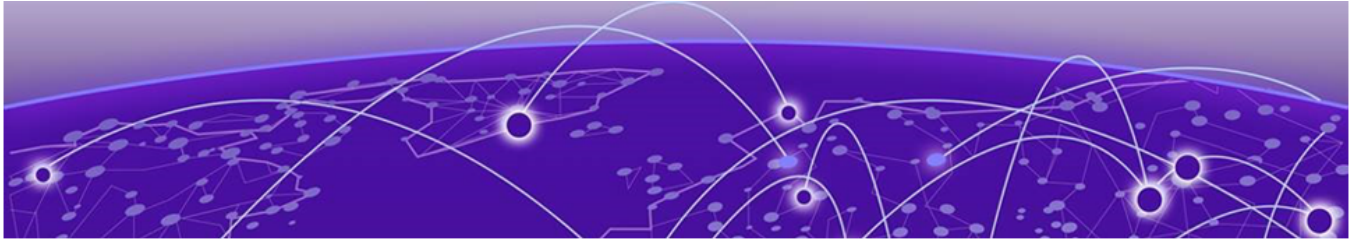
Table/Group	Supported Variables	Comments
	entConfigChange	An entConfigChange notification is generated when the value of entLastChangeTime changes. It can be utilized by an NMS to trigger logical/physical entity table maintenance polls. An agent should not generate more than one entConfigChange 'notification-event' in a given time interval (five seconds is the suggested default). A 'notification-event' is the transmission of a single trap or inform PDU to a list of notification destinations. If additional configuration changes occur within the throttling period, then notification-events for these changes should be suppressed by the agent until the current throttling period expires. At the end of a throttling period, one notification-event should be generated if any configuration changes occurred since the start of the throttling period. In such a case, another throttling period is started right away. An NMS should periodically check the value of entLastChangeTime to detect any missed entConfigChange notification-events.
entityGeneral	entLastChangeTime	The value of sysUpTime at the time a conceptual row is created, modified, or deleted in any of these tables: • entPhysicalTable • entLogicalTable • entLPMappingTable • entAliasMappingTable • entPhysicalContainsTable

VPLS-MIB (draft-ietf-l2vpn-vpls-mib-02.txt)

The following tables, groups, and variables are supported in this MIB.

All tables and variables of this MIB are supported as read only. The comments here are abbreviated versions of the description in the RFC documentation.

Table/Group	Supported Variables	Comments
vplsConfigTable	vplsConfigIndex	Unique index for the conceptual row identifying a VPLS service.
	vplsConfigName	A textual name of the VPLS. If there is no local name, or this object is otherwise not applicable, then this object MUST contain a zero-length octet string.
	vplsConfigAdminStatus	The desired administrative state of the VPLS service.
	vplsConfigRowStatus	For creating, modifying, and deleting this row.
	vplsConfigMtu	The value of this object specifies the MTU of this vpls instance.
	vplsConfigVpnId	This object indicates the IEEE 802-1990 VPN ID of the associated VPLS service.
vplsStatusTable	All objects	This table provides information for monitoring VPLS.
vplsPwBindTable	vplsConfigIndex	This table provides an association between a VPLS service and the corresponding PWs. A service can have more than one PW association. PWs are defined in the pwTable.
	vplsPwBindIndex	
	vplsPwBindType	The value of this object indicates whether the PW binding is of type mesh or spoke.
	vplsPwBindRowStatus	For creating, modifying, and deleting this row.



Troubleshooting

Troubleshooting Checklists on page 2107
LEDs on page 2108
Cable Diagnostics on page 2109
Using the Command Line Interface on page 2111
Using ELRP to Perform Loop Tests on page 2118
Using Simple Loop Prevention Protocol (SLPP) Guard on page 2125
Debug Mode on page 2127
Saving Debug Information on page 2128
Evaluation Precedence for ACLs on page 2130
TOP Command on page 2130
TFTP Server Requirements on page 2131
System Odometer on page 2131
Temperature Operating Range on page 2131
Understanding the Error Reading Diagnostics Message on page 2132
Proactive Tech Support on page 2132
Technical Support Bundles for Troubleshooting on page 2133

If you encounter problems when using the switch, this appendix may be helpful. Visit the Extreme Networks Support portal at https://extremeportal.force.com/ if you have a problem not listed here or in the release notes,.

Troubleshooting Checklists

This section provides simple troubleshooting checklists for Layer 1, Layer 2, and Layer 3. The commands and recommendations described are applicable to both IPv4 and IPv6 environments unless otherwise specified. If more detailed information about a topic is available, you are referred to the applicable section in this appendix.

Layer 1

When troubleshooting Layer 1 issues, verify:

- The installation of cables and connectors.
- The behavior of LED status lights. For additional information about LEDs, see LEDs.

- That the port is enabled, the link status is active, and speed and duplex parameters match the port settings at the other end of the cable.
  a. To display the configuration of one or more ports, use the `show ports configuration` command.
- That the packets are being received and transmitted.
  a. To display the number of packets being received and transmitted, use the `show ports {port_list | stack-ports stacking-port-list} statistics {no-refresh}` command.
- That there are no packet errors.
  a. To display packet error statistics, use the following commands:

  `show ports {port_list | stack-ports stacking-port-list} rxerrors {no-refresh}`—Displays receive error statistics

  `show ports {port_list | stack-ports stacking-port-list} txerrors {no-refresh}`—Displays transmit error statistics

  `show ports {mgmt | port_list |tag tag} collisions {no-refresh}`—Displays collision statistics

Layer 2

Layer 2 is the second, or data link, layer of the OSI model, or the MAC layer. This layer is responsible for transmitting frames across the physical link by reading the hardware, or MAC, source and destination addresses.

Layer 3

Layer 3 is the third layer of the OSI model. Also known as the network layer, Layer 3 is responsible for routing packets to different LANs by reading the network address.

LEDs

Power LED does not light:

Check that the power cable is firmly connected to the device and to the supply outlet.

On powering-up, the MGMT LED lights yellow:

The device has failed its Power On Self Test (POST) and you should contact your supplier for advice.

A link is connected, but the Status LED does not light:

Check that:

- All connections are secure.
- Cables are free from damage.

- The devices at both ends of the link are powered-up.
- Both ends of the Gigabit link are set to the same autonegotiation state.

  The Gigabit link must be enabled or disabled on both sides. If the two sides are different, typically the side with autonegotiation disabled will have the link LED lit, and the side with autonegotiation enabled will not be lit. The default configuration for a Gigabit port is autonegotiation enabled. Verify by entering the following command:

  ```
  show ports configuration
  ```

Predictive Failure LED on the AC power supply blinks amber:

Check the current status of the power supply. If the speed of both fans is above 2000 RPM, the AC power supply unit (PSU) is operating normally and no failure is imminent. To check and view the health of the installed PSU, use the following command:

```
show power {ps_num} {detail}
```

Switch does not power up:

All products manufactured by Extreme Networks use digital power supplies with surge protection. In the event of a power surge, the protection circuits shut down the power supply.

To reset the power, unplug the switch for 1 minute, plug it back in, and attempt to power-up the switch. If this does not work, try using a different power source (different power strip/outlet) and power cord.

Cable Diagnostics

The **`run diagnostics cable ports`** command enables cable diagnostics on copper ports using Time Domain Reflectometry (TDR) technology embedded in the PHY chips. This diagnostic tool helps identify physical issues such as cable breaks, shorts, or interference on individual twisted pairs.

Note
Running cable diagnostics may momentarily interfere with traffic on active ports.

Supported Ports

Cable diagnostics are supported **only on copper ports**. The command is not supported on:

- Extended BPE ports
- Stacking ports
- Fiber ports

Diagnostic Result States

State	Meaning
`Ok`	No fault detected; pair is functioning normally.
`Open`	Pair is not terminated or has a physical break. May be expected if unplugged.
`Short`	Conductors are shorted together.
`Crosstalk`	Interference detected between pairs; may indicate poor shielding or cable quality.
`Unknown`	Diagnostic failed or result is inconclusive.

Note
An `Open` state may be expected if the cable is not connected at the far end. If the cable is expected to be connected, this may indicate a fault.

Interpreting "Good" states:

- **Gigabit Ethernet (1000BASE-T)**: All four pairs should report `Ok`.
- **Fast Ethernet (100BASE-TX)**: Only pairs A and B are used; pairs C and D may show `Open` and still be valid.

Example

The following example runs diagnostics on port 14:

```
 # run diagnostics cable ports 14
Do you want to continue cable diagnostics? (y/N) Yes
cablediag: 1.1.0.14


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                             C A U T I O N                                  +
+   cablediag will momentarily interfere with traffic on active ports        +
+-------------------------------------------------------------------------+
+   Ports: 14                                                                +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Collecting port cable diagnostic information may take a moment...
port 14: cable (4 pairs, length +/- 10 meters)
        pair A Ok, length 0 meters
        pair B Ok, length 0 meters
        pair C Ok, length 4 meters
        pair D Ok, length 4 meters
```

Example with Faults

The following example runs diagnostics on port 3, which returns faults:

```
port 3: cable (4 pairs, length +/- 10 meters)
pair A Ok, length 5 meters
pair B Open, length 7 meters
pair C Crosstalk, length 6 meters
pair D Ok, length 5 meters
```

Best Practices

- Run diagnostics during maintenance windows to avoid traffic disruption.
- Use results to validate cabling during installation or troubleshooting.
- Document results for future reference and comparison.

Using the Command Line Interface

This section describes helpful information for using and understanding the command line interface (CLI).

General Tips and Recommendations

The initial welcome prompt does not appear:

Check that:

- Your terminal or terminal emulator is correctly configured
- Your terminal or terminal emulator has the correct settings:
  - 115200 baud for all ExtremeSwitching Universal switches
  - 8 data bits
  - 1 stop bit
  - no parity
  - XON/OFF flow control enabled

For console port access, you might need to press [Return] several times before the welcome prompt appears.

The SNMP Network Manager cannot access the device:

Check that:

- The *SNMP (Simple Network Management Protocol)* access is enabled for the system.
- The device IP address, subnet mask, and default router are correctly configured, and that the device has been reset.
- The device IP address is correctly recorded by the SNMP Network Manager (refer to the user documentation for the Network Manager).
- The community strings configured for the system and Network Manager are the same.
- The SNMPv3 USM, Auth, and VACM configured for the system and Network Manager are the same.

The Telnet workstation cannot access the device:

Check that:

- The device IP address, subnet mask, and default router are correctly configured, and that the device has been reset.
- You entered the IP address of the switch correctly when invoking the Telnet facility.
- Telnet access is enabled for the switch.

If you attempt to log in and the maximum number of Telnet sessions are being used, you should receive an error message indicating so.

Traps are not received by the SNMP Network Manager:

Check that the SNMP Network Manager's IP address and community string are correctly configured, and that the IP address of the Trap Receiver is configured properly on the system.

The SNMP Network Manager or Telnet workstation can no longer access the device:

Check that:

- Telnet access or SNMP access is enabled for the system.
- The port through which you are trying to access the device has not been disabled. If it is enabled, check the connections and network cabling at the port.
- The port through which you are trying to access the device is in a correctly configured *VLAN (Virtual LAN)*.
- The community strings configured for the device and the Network Manager are the same.

Try accessing the device through a different port. If you can now access the device, a problem with the original port is indicated. Re-examine the connections and cabling.

A network problem might be preventing you from accessing the device over the network. Try accessing the device through the console port.

Permanent entries remain in the FDB:

If you have made a permanent entry in the *FDB (forwarding database)* that requires you to specify the VLAN to which the entry belongs and then deleted the VLAN, the FDB entry remains. Although this does not harm the system, if you want to removed the entry, you must manually delete it from the FDB.

Default and static routes:

If you have defined static or default routes, those routes remain in the configuration independent of whether the VLAN and VLAN IP address that used them remains. You should manually delete the routes if no VLAN IP address is capable of using them.

You forget your password and cannot log in:

If you are not an administrator, another user having administrator access level can log in, delete your user name, and create a new user name for you, with a new password.

Alternatively, another user having administrator access level can log in and initialize the device. This will return all configuration information (including passwords) to the initial values.

In the case where no one knows a password for an administrator level user, contact your supplier.

The ExtremeSwitching switch displays only the "(pending-AAA) login" prompt

It is possible that the switch has not yet been fully initialized in the SummitStack. Wait for the Authentication Service (AAA) on the master node is now available for login. message to appear and then log in normally.

It is also possible that the stack has no master-capable node. In this case, AAA authentication will not be possible. This can occur when a stack has been deliberately dismantled and stacking was not unconfigured beforehand.

In either case, you may log in to the node using the failsafe account. If you have forgotten this account, and the stack has no master-capable node or has been dismantled, see Rescuing a Stack that has No Primary-Capable Node on page 216.

The "(Pending-AAA) login:" prompt on other switches.

This login prompt is discussed in Logging in to the Switch on page 31.

Node Prompt—SummitStack Only

You do not know which Node you are connected to:

If you use a console connection to access and configure the switch, you should connect to the console port of the master node.

The word "Slot" followed by a hyphen and a single decimal digit slot number will be inserted into the prompt in stacking mode. A sample of this prompt is:

```
* Slot-6 Stack.21 #
```

The sample indicates a changed configuration (*), the stackable is in stacking mode and is currently using the slot number 6 in the active topology ("Slot-6"), the system name is the default of "Stack", the command about to be executed is the 21st command, and the user is logged in as the administrator on the Master node (#).

There will be no specific prompt that indicates the node role. Run show switch command to discover the identities of the Master and Backup nodes. A successful login on a Standby node will show the ">" character instead of the "#" character at the end of the prompt.

Command Prompt

You do not know if the switch configuration has been saved:

If an asterisk (*) precedes the command prompt, a new change to the switch configuration has not been saved. To save the configuration, use the `save configuration` command. After you save the configuration, the asterisk (*) no longer precedes the command prompt.

You do not know if you are logged in as an administrator or a user:

Observe the console prompt. If you are logged in as an administrator, the prompt ends with the hash symbol (#). If you are logged in as a user, the prompt ends with a greater than sign (>).

The following is sample output from an administrator-level account:

```
BD-10808.1 #
```

The following is sample output from a user-level account:

```
BD-10808.1 >
```

Port Configuration

No link light on 10/100 Base port:

If patching from a switch to another switch, ensure that you are using a category 5 (CAT5) crossover cable. This is a CAT5 cable that has pins 1 and 2 on one end connected to pins 3 and 6 on the other end.

Excessive RX CRC errors:

When a device that has autonegotiation disabled is connected to an Extreme Networks switch with autonegotiation enabled, the Extreme Networks switch links at the correct speed, but in half-duplex mode. The Extreme Networks switch 10/100 physical interface uses a method called parallel detection to bring up the link. Because the other network device is not participating in autonegotiation (and does not advertise its capabilities), parallel detection on the Extreme Networks switch is able only to sense 10 Mbps versus 100 Mbps speed and not the duplex mode. Therefore, the switch establishes the link in half-duplex mode using the correct speed.

The only way to establish a full-duplex link is either to force it at both sides, or run autonegotiation on both sides (using full-duplex as an advertised capability, which is the default setting on the Extreme Networks switch).

Note

A mismatch of duplex mode between the Extreme switch and another network device causes poor network performance. Viewing statistics using the `show ports rxerrors` command on the Extreme Networks switch may display a constant increment of CRC errors. This is characteristic of a duplex mismatch between devices. This is NOT a problem with the Extreme Networks switch.

Always verify that the Extreme Networks switch and the network device match in configuration for speed and duplex.

No link light on Gigabit fiber port:

Check that:

- The transmit fiber goes to the receive fiber side of the other device and vice-versa. All Gigabit fiber cables are of the crossover type.
- The Gigabit ports are set to Auto Off (using the command `configure ports port_list {medium [copper | fiber]} auto off speed speed duplex [half | full]`) if you are connecting the Extreme Networks switch to devices that do not support autonegotiation.

  By default, the Extreme Networks switch has autonegotiation set to On for Gigabit ports and set to Off for 10 Gigabit ports.
- You are using multimode fiber (MMF) when using a 1000BASE-SX small form-factor pluggable (SFP), and single-mode fiber (SMF) when using a 1000BASE-LX SFP. 1000BASE-SX technology does not work with SMF. The 1000BASE-LX technology works with MMF but requires the use of a mode conditioning patchcord (MCP).

Software License Error Messages

You do not have the required software license:

If you attempt to execute a command and you do not have the required license, the switch returns the following message:

```
Error: This command cannot be executed at the current license level.
```

You have reached the limits defined by the current software license level:

If you attempt to execute a command and you have reached the limits defined by the current license level the switch returns the following message:

```
Error: You have reached the maximum limit for this feature at this license level.
```

For more information about licensing requirements, see the Switch Engine v33.6.1 Licensing Guide document.

VLANs

You cannot add a port to a VLAN:

If you attempt to add a port to a and get an error message similar to:

```
localhost:7 # configure vlan marketing add ports 1:1,1:2
Error: Protocol conflict when adding untagged port 1:1. Either add this port as tagged or
assign another protocol to this VLAN.
```

You already have a VLAN using untagged traffic on a port. Only one VLAN using untagged traffic can be configured on a single physical port.

You verify the VLAN configuration using the following command:

show vlan

The solution for this error using this example is to remove ports 1 and 2 from the VLAN currently using untagged traffic on those ports.

If this were the "default" VLAN, the command would be:

```
localhost:23 # configure vlan default delete ports 1:1,1:2
```

You can now re-enter the previous command without error:

```
localhost:26 # configure vlan marketing add ports 1:1,1:2
```

VLAN names:

There are restrictions on VLAN names. They cannot contain whitespaces and cannot start with a numeric value.

VLANs, IP addresses, and default routes:

The system can have an IP address for each configured VLAN. You must configure an IP address associated with a VLAN if you intend to manage (Telnet, , ping) through that VLAN or route IP traffic.

You can also configure multiple default routes for the system. The system first tries the default route with the lowest cost metric.

STP

You have connected an endstation directly to the switch and the endstation fails to boot correctly:

The switch has the enabled, and the endstation is booting before the STP initialization process is complete. Specify that STP has been disabled for that , or turn off STP for the switch ports of the endstation and devices to which it is attempting to connect; then, reboot the endstation.

Spanning Tree Domain names:

There are restrictions on names. They cannot contain whitespaces and cannot start with a numeric value.

You cannot add ports within a VLAN to the specified STPD:

Check to ensure that you are adding ports that already exist in the carrier VLAN.

If you see an error similar to the following:

```
Error: Cannot add VLAN default port 3:5 to STP domain.
```

You might be attempting to add:

- Another 802.1D mode STP port to a physical port that already contains an 802.1D mode STP port (only one 802.1D encapsulation STP port can be configured on a particular STP port).
- A carrier VLAN port to a different STP domain than the carrier VLAN belongs.

- A VLAN and/or port for which the carrier VLAN does not yet belong.

Note
This restriction is only enforced in an active STPD and when you enable STP to make sure you have a legal STP configuration.

Only one carrier VLAN can exist in an STPD:

Only one carrier VLAN can exist in a given STPD although some of the ports on the carrier VLAN can be outside the control of any STPD at the same time.

The StpdID must be identical to the VLANid of the carrier VLAN in that STPD.

The switch keeps aging out endstation entries in the switch FDB:

If the switch continues to age out endstation entries in the switch :

- Reduce the number of topology changes by disabling STP on those systems that do not use redundant paths.
- Specify that the endstation entries are static or permanent.

ESRP

ESRP names:

There are restrictions on Extreme Standby Router Protocol (ESRP) names. They cannot contain whitespaces and cannot start with a numeric value.

You cannot enable an ESRP domain:

Before you enable a specific ESRP domain, it must have a domain ID. A domain ID is either a user-configured number or the 802.1Q tag (VLANid) of the tagged master VLAN. The domain ID must be identical on all switches participating in ESRP for that particular domain. If you do not have a domain ID, you cannot enable ESRP on that domain.

Note the following on the interaction of tagging, ESRP, and ESRP domain IDs:

- If you have an untagged Master VLAN, you must specify an ESRP domain ID.
- If you have a tagged master VLAN, ESRP uses the 802.1Q tag (VLANid) of the master VLAN for the ESRP domain ID. If you do not use the VLANid as the domain ID, you must specify a different domain ID.

You cannot delete the master VLAN from the ESRP domain:

If you attempt to remove the master VLAN before disabling the ESRP domain, you see an error message similar to the following:

```
ERROR: Failed to delete master vlan for domain "esrp1" ; ESRP is enabled!
```

If this happens:

- Disable the ESRP domain using the disable esrp command.

- Remove the master VLAN from the ESRP domain using the configure esrp delete master command.

VRRP

You cannot define VRRP virtual router parameters:

Before configuring any parameters for VRRP, you must first create the VRRP instance on the switch.

If you define VRRP parameters before creating the VRRP, you may see an error similar to the following:

```
Error: VRRP VR for vlan vrrp1, vrid 1 does not exist.
Please create the VRRP VR before assigning parameters.
Configuration failed on backup MSM, command execution aborted!
```

If this happens:

- Create a VRRP instance using the `create vrrp vlan vrid` command.
- Configure the VRRP instance's parameters.

Using ELRP to Perform Loop Tests

The Extreme Loop Recovery Protocol (ELRP) is used to detect network loops in a Layer 2 network. A switch running ELRP transmits multicast packets with a special MAC destination address out of some or all of the ports belonging to a *VLAN*. All of the other switches in the network treat this packet as a regular, multicast packet and flood it to all of the ports belonging to the VLAN. When the packets transmitted by a switch are received back by that switch, this indicates a loop in the Layer 2 network.

Note
The ELRP source MAC address is changed from "00:e0:2b:00:00:01" to “0e:Switch-MAC”.

After a loop is detected through ELRP, different actions can be taken such as blocking certain ports to prevent loop or logging a message to system log. The action taken is largely dependent on the protocol using ELRP to detect loops in the network.

You can use ELRP on a “standalone” basis or with other protocols such as *ESRP (Extreme Standby Router Protocol)*, as described in Using ELRP with ESRP. Protocols such as *EAPS (Extreme Automatic Protection Switching)* require that a network have a ring topology to operate. In this case you can use ELRP to ensure that the network has a ring topology.

The “standalone” ELRP commands determine whether or not a network has an Layer 2 loop.

About Standalone ELRP

Standalone ELRP gives you the ability to send ELRP packets, either periodically or on an ad hoc “one-shot” basis on a specified subset of *VLAN* ports. If any of these transmitted packets is received back then standalone ELRP can perform a configured action such as sending a log message to the system log file, sending a trap to the *SNMP* manager, and disabling the port where the looped packet arrived.

ELRP uses QP8 to send/receive control packets over the network to detect loops, so it is advisable to not use QP8 in for user traffic in the network.

Standalone ELRP allows you to:

- Configure ELRP packet transmission on specified VLANs.
- Specify some or all the ports of the VLAN for packet transmission. Each VLAN must be configured individually for ELRP.

Note
Reception of packets is not limited to any specific ports of the VLAN and cannot be configured.

- Save and restore standalone ELRP configuration across reboots.
- Request non-periodic or periodic transmission of ELRP packets on specified ports of a VLAN.

Note
Starting with ExtremeXOS 22.2, ELRP works on both static and dynamic VLANs.

Non-periodic ELRP Requests

You can specify the number of times ELRP packets must be transmitted and the interval between consecutive transmissions. A message is printed to the console and logged into the system log file indicating detection of network loop when ELRP packets are received back or no packets are received within the specified duration.

Periodic ELRP Requests

You can configure the interval between consecutive transmissions. If ELRP packets are received back, a message is printed to the system log file and/or a trap is sent to the *SNMP* manager indicating detection of a network loop.

You have the option to configure the switch to automatically disable the port where the looped packet arrived as well as the length of time (in seconds) that the port should remain disabled. When this hold time expires, the port is automatically enabled.

Should a loop occur on multiple ports, only the first port in the *VLAN* on which the PDU is received is disabled. The second port is ignored for 1 or 2 seconds and then if another PDU is received, that port is disabled until the loop is gone. This prevents shutting down all ports in the VLAN.

ELRP Egress Port Disable

When a loop is detected via ELRP, an option to disable the port where the packet egressed was added to suppress the loop in ExtremeXOS 16.1. All existing functionalities supported on disabling the port where the packet ingressed are supported on this enhancement as well. A log message or *SNMP* trap will be generated based on the configuration. The user must specify a holdtime or keep the port disabled until a user enables it.

ELRP egress blocking enhancement adds the following value:

- More efficient bandwidth utilization.
- Better loop protection in environment. If an edge device is dual-homed to an MLAG pair and this edge device loses its *LAG (Link Aggregation Group)* configuration, ELRP will be able to prevent the loop by disabling egress port.

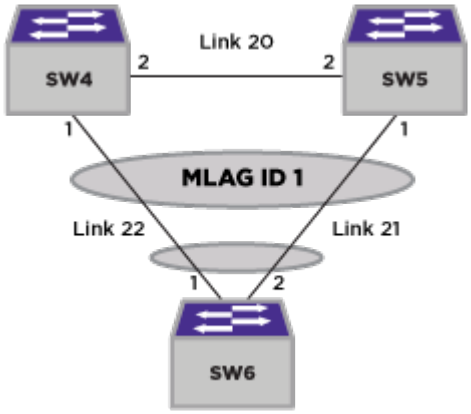


Figure 233: ELRP Egress Port Disable (1)

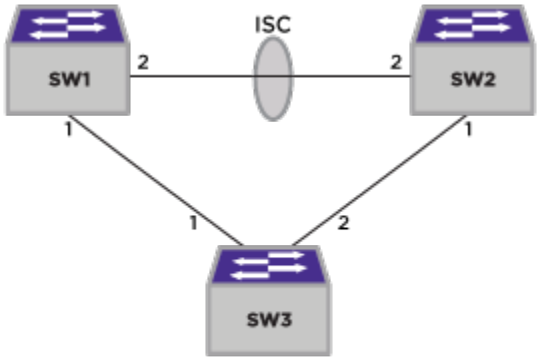


Figure 234: ELRP Egress Port Disable (2)

Exclude Port List

When you have configured the switch to automatically disable the port where the looped packet arrived, there may be certain ports that you do not want disabled. You can then create a list of ports that are excluded from this automatic disabling and that remain enabled. This list can also contain *EAPS* ring ports. When this option is selected, the actual EAPS ring ports do not have to be explicitly listed. You can also specify that

VXLAN (Virtual Extensible LAN) remote end points are to be excluded from getting disabled when a loop is detected.

You can configure any port on the switch into the exclude port list. The list can also be edited to add or delete any port on the switch. Then when ELRP detects a loop and the disable feature is enabled, it checks to see if the port from which the PDU arrived is on the user defined exclude port list or is part of a trunk port that is defined on the list. When the port is on the list, ELRP logs the event and does not disable the port. When the port is not on the list, ELRP disables the port.

Limitations

The following are limitations to this feature:

- A specified port is added to the list regardless of its corresponding *VLAN*.
- Only ports on the local switch can be added.
- A loop detected on an excluded port may persist indefinitely until user action is taken.
- ISC links should not be blocked, therefore they should be added to the excluded port list.

ELRP with Dynamic VLANs

ELRP supports dynamically created VLANs created by:

- *NetLogin*
- *Extreme Network Virtualization (XNV)*
- Multiple Registration Protocol (MVRP)
- Fabric Attach (starting with version 32.2)
- ONEPolicy (starting with version 32.2)

ELRP on NetLogin-enabled ports is supported.

ELRP with Virtual Extensible LAN (VXLAN)

Starting with ExtremeXOS 22.4, ELRP is supported on Virtual Extensible LAN (VXLAN).

VXLAN is an encapsulation mechanism. A VXLAN header is added to an L2 (ELRP PDU) frame and encapsulated in a UDP/IP packet. Transit nodes see only an ordinary IP packet. For more information about VXLAN, see VXLAN on page 2305.

ELRP detects loops across VXLAN tunnels. If a loop is detected across the tunnel and the disable-port option is configured (see Configuring Periodic Requests on page 2123), ELRP takes down the VXLAN remote endpoint.

If the VXLAN remote endpoint has been disabled using the permanent option, you can use the following command to re-enable the remote endpoint:

```
enable virtual-network remote-endpoint vxlan [ ipaddress ipaddress {vr
vr_name} | all ]
```

To manually disable a remote endpoint, use the following command:

```
disable virtual-network remote-endpoint vxlan [ ipaddress ipaddress {vr
vr_name} | all ]
```

To see if a remote endpoint is enabled or disabled, use the following command:

```
show virtual-network {vn_name} remote-endpoint vxlan {vni vni}
{ipaddress ipaddress {vr vr_name}}
```

Hardware-Assisted ELRP

Hardware-assisted ELRP allows ELRP to detect loops faster and to support more ELRP clients.

Supported Platforms

All ExtremeSwitching Universal switches.

Limitations

For hardware-assisted ELRP:

- Only front panel ports can be configured as the loopback port.
- Only disabling ingress port is supported after a loop is detected.
- MLAG ports are not supported.
- Software ELRP and hardware-assisted ELRP cannot operate simultaneously.
- Only standalone clients are supported.
- Not supported on dynamic VLANs.
- Maximum loop detection interval is determined by each platforms hardware capability. For most platforms, an ELRP packet is 70 bytes.
- Non-periodic requests are not supported.
- Extended Edge Switching is not supported.

Configuring Standalone ELRP

The ELRP client (standalone ELRP) must be enabled globally in order for it to work on any *VLAN*.

- Globally enable the ELRP client for static VLANs:

  ```
  enable elrp-client {software | hardware-assist}
  ```

  The ELRP client can be disabled globally so that none of the ELRP VLAN configurations take effect.
- Globally disable the ELRP client for static VLANs:

  ```
  disable elrp-client
  ```
- Globally enable or disable ELRP for dynamic VLANs:

  ```
  configure elrp-client dynamic-vlans {mvrp | netlogin | vm-tracking |
  policy | fabric-attach | all} [on | off]
  ```

- Globally enable or disable inter-VLAN loop detection:

  ```
  configure elrp-client inter-vlan-loop-detection [on | off]
  ```

- Configure or unconfigure a front panel port as the designated loopback port for hardware-assisted ELRP:

  ```
  configure elrp-client hardware-assist loopback-port [port | none]
  ```

  Note
  Do not configure a trunk port as the loopback port. In a stack, make sure the loopback port is not on the master slot.

Configuring Non-periodic Requests

- To start one-time, non-periodic ELRP packet transmission on specified ports of a *VLAN* using a particular count and interval, use one of the following commands:

  ```
  configure elrp-client one-shot {vlan [vlan_name | all} ports [ports |
  all |none] {remote-endpoints vxlan all} {interval interval {seconds |
  milliseconds}} {retry count} {log | print | print-and-log]}
  ```

  ```
  run elrp {vlan}vlan_name {ports [ports |all | none]} {remote-endpoints
  vxlan all} {interval interval {seconds | milliseconds} } {retry count}
  ```

  These commands start one-time, non-periodic ELRP packet transmission on the specified ports of the VLAN using the specified count and interval. If any of these transmitted packets is returned, indicating loopback detection, the ELRP client performs the configured actions of logging a message in the system log file and/or printing a log message to the console. There is no need to trap to the *SNMP* manager for non-periodic requests.

Configuring Periodic Requests

- To start periodic ELRP packet transmission on specified ports of a *VLAN* using a particular interval, use the following command:

  ```
  configure elrp-client periodic {vlan} vlan_name ports [ports | all
  | none] {remote-endpoints vxlan all} {interval interval {seconds |
  milliseconds}} {log | log-and-trap | trap} {disable-port {egress |
  ingress} {duration {seconds} | permanent}}
  ```

  This command starts periodic ELRP packet transmission on the specified ports of the VLAN using a specified interval. If any of these transmitted packets is returned, indicating loopback detection, the ELRP client performs the configured action of logging a message in the system log file and/or sending a trap to the *SNMP* manager.

  When the option of disabling a port is configured, you choose the duration in seconds as a hold time, or you disable the port permanently. When ELRP disables

the port, the operation is not persistent. When the switch is rebooted, the port is enabled when the switch comes up.

Note

ELRP detects loops on a per VLAN basis. When you select the disable-port option keep in mind that the entire port is disabled. In the case of *VXLAN*, the remote tunnel endpoint's (RTEP) state is taken down for all VXLAN virtual networks. This may affect connectivity for other VLANs configured on that port that did not have any data loop problems. ELRP also does not distinguish between uplink ports and host ports. When the **disable port** option is selected and ELRP detects a loop, any and all ports where the loop was detected will be disabled, including uplink ports.

- To disable a pending one-shot or periodic ELRP request for a specified VLAN or all dynamic VLANs, use the following command:

  `unconfigure elrp-client {vlan_name | dynamic-vlans}`

Configuring Exclude Port List

- To configure an ELRP exclude port list, use the following command:

  `configure elrp-client disable-ports [exclude | include] [ ports | eaps-ring-ports | remote-endpoints vxlan | inter-vlan-loop]`
- To unconfigure an ELRP exclude port list, use the following command:

  `unconfigure elrp-client disable-ports`

Displaying Standalone ELRP Information

- To display summary ELRP status information, use the following command:

  `show elrp`
- To display information about ELRP disabled ports, use the following comamnd:

  `show elrp disabled-ports`
- To display information ELRP configuration on dynamic VLANs, use the following command:

  `show elrp dynamic-vlans {vm-tracking | netlogin | mvrp | policy | fabric-attach | all}`

Example: ELRP on Protocol-Based VLANs

For ELRP to detect loops on a protocol-based VLAN (other than the protocol any), you need to add the ethertype 0x00bb to the protocol.

1. Create VLANs:

   ```
   create vlan v1
   create vlan v2
   ```
2. Configure protocol filters:

   ```
   configure vlan v1 protocol IP
   configure vlan v2 protocol decnet
   ```

3. Add ports to the VLANs:
```
configure vlan v1 add ports 1
configure vlan v2 add ports 2
```
4. Enable ELRP on the created VLANs:
```
enable elrp-client
configure elrp-client periodic v1 ports all interval 5 seconds log
configure elrp-client periodic v2 ports all interval 5 seconds log
```
5. Add the ethertype to the protocol:
```
configure protocol IP add snap 0x00bb
configure protocol decnet add snap 0x00bb
```

VLANs v1 and v2 can now detect the loop on their respective broadcast domains.

Using Simple Loop Prevention Protocol (SLPP) Guard

Simple Loop Prevention Protocol (SLPP) is an application that detects loops in an or Split Multi-link Trunking (SMLT) network on VOSS/BOSS switches.

SLPP Guard is a complementary feature for ExtremeXOS switches that helps prevent loops in networks by administratively disabling an edge port if a switch receiveS an SLPP PDU from an SMLT network. SLPP Guard detects and discards SLPP control PDUs on a per port basis. SLPP Guard identifies SLPP PDUs using the Ethernet type field of the packet, which is configurable. When an SLPP PDU is received on a port that has SLPP Guard activated on it, it is immediately disabled. After a configurable timeout value expires (associated with each port), the port is automatically re-enabled.

Note
When SLPP Guard disables a port, the disabled status is not persistent. If the switch reboots, the port is enabled when the switch comes back up.

Note
If you use the CLI to enable or disable a port that has been disabled by SLPP Guard, the port is enabled or disabled as applicable and the recovery timer is stopped for that port.

When a new user is authenticated by netlogin, the *enable* attribute of SLPP Guard is given to netlogin. Netlogin processes this attribute and sends the *enable* attribute message to the SLPP Guard module. If all users are unauthenticated by netlogin, a disable message for the port is sent to SLPP Guard. At the time of re-authentication, if the FA-Service-Request = SLPPGUARD is removed from the RADIUS server, then netlogin sends an SLPP Guard *disable* message.

Note
If you modify the RADIUS attributes in between authentication, then a port restart or clear all netlogin clients is required in order for the service attributes to change on the port.

The status is enabled and the entry in the **save configuration** command is sent to `enable slpp guard port` *port_list* for the following scenarios:

- If the status of SLPP Guard is enabled by the CLI.

- If the status of SLPP Guard is enabled by both the Vendor Specific Attribute (VSA) and by the CLI.
- If after enabling SLPP Guard by the CLI, the status is then disabled from the VSA.

The status is enabled and the saved configuration contains default values for the following scenarios:

- If the status of SLPP Guard is enabled by the VSA.
- If after enabling SLPP Guard by the VSA, the status is then disabled from the CLI.

The most common use case is with an IP phone with two ports: one for a switch and one for a computer (see Figure 235). If you plug both IP phone ports onto the edge switch, a loop is created. To avoid a loop in the SMLT network, on the ExtremeXOS edge switch, enable SLPP Guard on the ports that are connected to the IP phone.

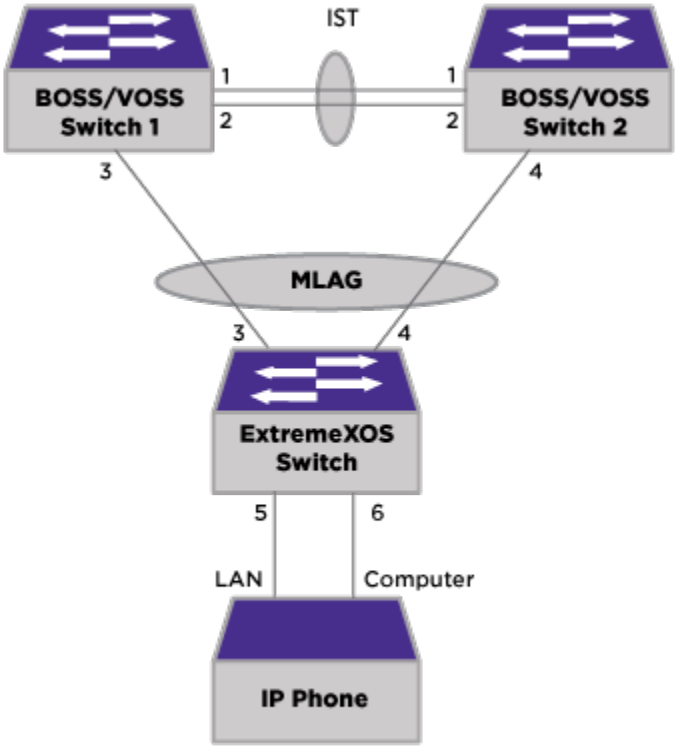


Figure 235: SLPP Use Case

Supported Platforms

All ExtremeSwitching switches.

Limitations

- SNMP for SLPP Guard is not supported.
- The SLPP Guard active state per port is limited by the number of available ACL filters.

Configuring Simple Loop Prevention Protocol (SLPP) Guard

For Simple Loop Prevention Protocol (SLPP) Guard to operate, SLPP must have already been configured in the core of the network.
The following information explains how to configure SLPP Guard on ExtremeXOS switches.

1. Enable SLPP on the desired ports using the following command:

   `enable` **`slpp guard ports`** [*port_list* | **`all`**]

   Note
   To view the SLPP status of ports, use the `show` **`slpp guard`** {**`ports`** *port_list*} {**`disabled-ports`**} command.

2. (Optional) Set the recovery timeout period using the following command:

   `configure` **`slpp guard`** [**`ports`** [*port_list* | **`all`**] **`recovery-timeout`** [*seconds* | **`none`**]

   After the configured timeout value set by this command expires (associated with each port), the port is automatically re-enabled.
3. (Optional) Configure the Ethertype for SLPP Guard using the following command:

   `configure` **`slpp guard ethertype`** *hex*

   This command configures the Ethertype that the SLPP Guard feature uses to identify SLPP PDUs.

If you need to disable SLPP Guard on a port, use the following command:

`disable` **`slpp guard ports`** [*port_list* | **`all`**]

To view SLPP Guard status for selected ports or all SLPP Guard-disabled ports, use the following command:

`show` **`slpp guard`** {**`ports`** *port_list*} {**`disabled-ports`**}

Debug Mode

The EMS provides a standard way to filter and store messages generated by the switch. With EMS, you must enable debug mode to display debug information. You must have administrator privileges to use these commands. If you do not have administrator privileges, the switch rejects the commands.

- Enable or disable debug mode for EMS:

  `enable log debug-mode`

After debug mode has been enabled, you can configure EMS to capture specific debug information from the switch. Details of EMS can be found in Status Monitoring and Statistics.

Saving Debug Information

You can save switch data and statistics to a network TFTP server, an internal memory card that comes pre-installed in the switch or a removable USB 2.0 storage device on all Universal platforms.

With assistance from Extreme Networks Technical Support personnel, you can configure the switch to capture troubleshooting information, such as a core dump file, to these locations.

The switch only generates core dump files in the following situations:

- If an ExtremeXOS process fails.
- When forced under the guidance of Extreme Networks Technical Support.

The core dump file contains a snapshot of the process when the error occurred.

Note

Use the commands described in this section only under the guidance of Extreme Networks Technical Support personnel to troubleshoot the switch.

Before you can enable and save process core dump information to removable storage devices, you must install a compact flash card or USB 2.0 storage device. .

Enabling the Send Debug Information Switch

This allows information to be sent to the internal memory card or a removable storage device.

- Enable the switch to save process core dump information.

  ```
  configure debug core-dumps [ off | directory_path]
  ```

  A removable storage device is a USB 2.0 storage device on a ExtremeSwitching series switch.

  Core dump files have a .gz file extension. The filename format is: core.<process-name.pid>.gz where process-name indicates the name of the process that failed and pid is the numerical identifier of that process.

  If you configure the switch to write core dump files to the internal memory card and attempt to download a new software image, you might have insufficient space to complete the image download.

  If this occurs, you must decide whether to continue the software download or move or delete the core dump files from the internal memory. For example, if your switch supports a removable storage device with space available, transfer the files to the storage device. On switches that do not have a removable storage device, transfer

the files from the internal memory card to a TFTP server. This frees up space on the internal memory card while keeping the core dump files.

Copy Debug Information to Removable Storage Devices

- Save and copy debug information to the specified USB 2.0 storage device.

  `save debug tracefiles` *`directory_path`*

  After the switch writes a core dump file or other debug information to the storage device, and before you can view the contents on the device, you must ensure it is safe to remove the device from the switch. Use the `eject` *`usb-device`* command to prepare the device for removal. After you run the `eject` *`usb-device`* command, you can manually remove the storage device from the switch and read the data on the device.

To access and read the data on a removable storage device, use a PC with the appropriate hardware, such as a compact flash reader/writer and follow the manufacturer's instructions to access the compact flash card and read the data.

Copying Debug Information to a TFTP Server

SummitStack only—To get debug information from a node in the SummitStack that is not a master node, you must configure an alternate IP address on the node and have its management port connected. You may then use the tftp command to upload specific files. The upload debug command functions only on the master node.

1. Save and copy debug information to the specified TFTP server.

   `upload debug [` *`ipaddress`* `|` *`hostname`* `] {{`**`vr`**`}` *`vrname`*`} {`**`block-size`** *`block_size`*`}`

   Progress messages are displayed that indicate the file being copied and when the copying is finished.

   Depending on your platform, the switch displays a message similar to the following:

   ```
   The following files on have been uploaded:
   Tarball Name: TechPubsLab_C_09271428.tgz
   ./primary.cfg
   ```

   You can also use this command in conjunction with the `show tech` command.

   Prior to uploading debug information files, the switch prompts you with the following message to run the `show tech-support` command with the logto file option:

   ```
   Do you want to run show tech logto file first? (y/n)
   ```

2. Enter y to run the `show tech-support` command before uploading debug information.

   If you enter y, the show_tech.log.tgz file is included during the upload. Enter n to upload debug information without running the `show tech` command.

   After you upload the debug information, you should see a compressed TAR file on the TFTP server, which contains the debug information.

   The TAR file naming convention is

```
<SysName>_<{<slot#>|A|B}I|C>_<Current Time>.tgz
- Current Time = mmddhhmm ( month(01-12), date(01-31), hour(0-24), minute(00-59) ).
```

Managing Debug Files

For the purposes of this section, it is assumed that you have configured the switch to send core dump information under the guidance of Extreme Networks Technical Support.

Managing the debug files might include any of the following tasks: renaming or copying a core dump file, displaying a comprehensive list of files including core dump files, transferring core dump files, and deleting a core dump file. You can manage the debug files using the same commands that you use to manage other switch files.

Note
Filenames are case-sensitive. For information on filename restrictions, refer to the specific command in the Switch Engine v33.6.1 Command References .

Evaluation Precedence for ACLs

The ACLs on a port are evaluated in the following order:

- Persistent dynamic ACLs
- Host-integrity permit ACLs
- MAC source address deny ACLs
- Source IP lockdown source IP permit ACLs
- Source IP lockdown deny all ACLs
- ARP validation CPU ACLs
- ACLs created using the CLI
- DoS Protect-installed ACLs
- MAC-in-MAC installed ACLs
- ACLs applied with a policy file (see ACLs on page 341 for precedence among these ACLs)

TOP Command

The `top` command is a UNIX-based command that displays real-time CPU utilization information by process.

The output contains a list of the most CPU-intensive tasks and can be sorted by CPU usage, memory usage, and run time. For more detailed information about the `top` command, refer to your UNIX documentation.

TFTP Server Requirements

We recommend using a TFTP server that supports blocksize negotiation (as described in RFC 2348, TFTP Blocksize Option), to enable faster file downloads and larger file downloads.

System Odometer

Each field replaceable component contains a system odometer counter in EEPROM.

The `show odometers` command displays an approximate days of service duration for an individual component since the component was manufactured.

Monitored Components

The odometer monitors the following components:

- Switch
- XGN-2xn card

Recorded Statistics

Description

The following odometer statistics are collected by the switch:

- Service Days—The amount of days that the component has been running
- First Recorded Start Date—The date that the component was powered-up and began running

Depending on the software version running on your switch, additional or different odometer information may appear.

The following is sample output from an ExtremeSwitching series switch:

```
# show odometers
Service  First Recorded
Field Replaceable Units           Days     Start Date
------------------------         -------  --------------
Switch     : SummitX4                   7  Dec-08-2004
XGM-2xn-1  :
```

Temperature Operating Range

On ExtremeSwitching series switches, if a switch runs outside the expected range, the switch logs an error message, generates a trap, and one of the following actions occurs:

- On a SummitStack slots, the reset/none option from `configure sys-recovery-level [all | none]` command determines the action that occurs.

- For all switches, only a warning is logged and the switch continues to run with the system state as "overheat." If the temperature increases further into overheat condition, after the operating temperature exceed a further higher temperature, the hardware resets the switch.

Understanding the Error Reading Diagnostics Message

For more detailed infomation about error messages and their meanings, see the Switch Engine and ExtremeXOS v33.6.x EMS Messages Catalog .

If you have never run diagnostics on the switch or stack ports and use the `show diagnostics` command, the switch displays a message similar to the following:

```
Result: FAIL
Test date run is invalid. Please run Diagnostics.
Error reading diagnostics information.
```

This message is normal and expected if you have never run diagnostics on the switch. After running diagnostics, you see information about the executed test using the `show diagnostics` command.

Proactive Tech Support

Proactive tech support functionality allows you to collect system information when support is required. System information is pushed into a cloud-hosted collector where the Extreme TAC can use it to identify problems to provide solutions. Proactive Tech Support provides the following functionality:

- Proactive tech support is implemented as a loadable application, called techSupport.xmod. The XMOD is bundled with ExtremeXOS image packages, but can also be upgraded independently at any time without restarting the switch.
- Proactive tech support allows the switch to proactively collect switch status and send the status report to up to four cloud-hosted collectors.
- Proactive tech support allows both automatic and manual status report.
- Proactive tech support allows each collector to have different report data set, report frequency, or report mode. A default collector is configured automatically to minimize the configuration required to enable the feature.

Limitations

- Proactive tech support uses SSL to secure the switch information transmission on the Internet. If the feature is not enabled, the switch information is transmitted as clear text.
- XMOD applications can be dynamically upgraded without restarting the switch . But the upgraded XMOD version needs to match the installed ExtremeXOS version.

Locating a Collector

The switch attempts to locate a data collector using the DNS hostname, or the IP address of a collector. You specify the DNS hostname or IP address of a collector when you add a collector. When a DNS hostname is configured for a collector, the switch attempts to resolve the IP address for the configured DNS hostname before it attempts to connect to the collector. If the process of resolving IP address for a collector fails, the connection process stops and logs an error message.

When you start the feature a default collector is automatically configured a public IP address of a Splunk server hosted in Extreme's internal network. When you enable the feature, the switch automatically attempts to locate the default collector and send switch status information to the collector. You can configure up to four collectors on the switch. If you want to collect switch status information in your own collector (beside the default collector), you can use the `configure tech-support add collector` command to configure an additional collector. The switch will send independent status reports to each collector based on each collector's configuration.

By default, the primary IP address on the Mgmt *VLAN* interface is used for the discovery process of a collector. If an IP address is not found on the Mgmt VLAN, an error messaged is displayed. Optionally, you can specify a source IP address to connect to a collector. The connection between the switch and a collector is established and maintained with TCP.

Configuring Proactive Tech Support

The examination and reporting of switch status is implemented as a loadable application, called techSupport.xmod. You can dynamcally upgrade the XMOD applications without restarting the switch by issuing `download image` *`ip_address app.xmod`*, `run update`, and then `restart process techSupport` commands.

Safe Default Startup

By default proactive tech support services are disabled. You can opt-out of this service either by answering "Yes", "Y", or pressing the ENTER key to answer the question that is posed when accessing the switch for the first time. You can also use the `disable tech-support` collector command. Here is the system output prompt for the tech support feature:

```
The switch will proactively attempt to send basic configuration and
operational switch information for the purpose of assisting Extreme
Networks resolve customer reported issues. Uploaded data is encrypted if
the ssh.xmod is installed. Otherwise, a reduced switch data set is sent
in clear text that contains no customer specific information.

Would you like to disable the automatic switch reporting service? [Y/n]:
```

Technical Support Bundles for Troubleshooting

To troubleshoot issues, you can use debug bundles, which are JSON files with predefined modules to common problems that can be executed when an issue occurs.

You can search for a bundle that best matches your issue, and then execute that relevant bundle that matches your problem.

To use tech support bundles:

1. View the available bundles using the following command:

   ```
   show tech-support help
   ```

   List of Bundles: L2-VPN, AAA, GPTP, MLAG, optics, VLAN, MPLS, L2 multicast, Netlogin , L3 IPv4, VRRP, LAG, L2, EAPS, policy, STP, ACL, OSPF, link issues, Extended Edge Switching (VPEX).
2. Copy `debug_bundle.json` to a TFTP server. Transfer by TFTP the file to your switch using the following command:

   `tftp get [ ` **`ip-address`** ` | ` **`host-name`**`] { ` **`vr`** *`vr_name`* ` } { ` **`block-size`** *`block_size`* ` } *remote-file local-file*} {`**`force-overwrite`**`}`

   Example:

   ```
   # tftp get 10.120.89.96 debug_bundle.json
   ```

   The `debug_bundle.json` file is located at: https://github.com/extremenetworks/ExtremeScripting/blob/master/EXOS/Debug_Bundle/debug_bundle.json
3. Run a debug bundle using the following command:

   `show tech-support {`**`all`** ` | *area* | {{`**`bundle`** *`bundle_name`*`} {`**`monitor`** *`num_iterations`*`} {`**`interval`** *`seconds`*`}} {`**`detail`**`} {`**`logto`** `[`**`file`**`]}`

Universal Port

Profile Types on page 2136
Dynamic Profile Trigger Types on page 2138
How Device-detect Profiles Work on page 2142
How User Authentication Profiles Work on page 2142
Profile Configuration Guidelines on page 2144
Collecting Information from Supplicants on page 2150
Supplicant Configuration Parameters on page 2151
Universal Port Configuration Overview on page 2151
Using Universal Port in an LDAP or Active Directory Environment on page 2153
Configuring Universal Port Profiles and Triggers on page 2154
Managing Profiles and Triggers on page 2156
Sample Universal Port Configurations on page 2159

Universal Port is a flexible framework that enables automatic switch configuration in response to special events such as:

- User login and logoff
- Device connection to or disconnection from a port
- Time of day
- EMS event messages

Note
The Universal Port feature is supported only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

The primary component of the Universal Port feature is the profile, which is a special form of command script that runs when triggered by the events mentioned above.

Profiles execute commands and use variables as do the scripts described in Using CLI Scripting on page 2182. The primary difference is that a profile can be executed manually or automatically in response to switch events.

Note
The term *profile* is distinct from the term *policy* because a policy is only one particular application of a profile.

Universal Port works with the following ExtremeXOS components and third-party products:

- ExtremeXOS Network Login
- ExtremeXOS LLDP
- ExtremeXOS CLI Scripting
- Status Monitoring and Statistics
- RADIUS Servers
- Active directory services such as LDAP and Microsoft Active Directory

The following are some examples of how you can use Universal Port on a network:

- Automatically provision a VoIP phone and the attached switch port with appropriate *PoE (Power over Ethernet)* budget and *QoS (Quality of Service)* settings when the phone connects.
- Create security policies that can follow a user as the user roams around a campus. For example, an engineer can walk from Building 1 to Building 5, plug his PC into the network and be authenticated with the appropriate access rights and ACLs.
- Support separate authentication for VoIP phones and workstations on the same port.
- Create profile templates with variables so that you can re-use templates with different address ranges and parameters.
- Apply different security policies for different locations (for example, a restricted area).
- Disable wireless access after business hours.

Note
Special scripts can be run when the switch boots. For more information, see Using Autoconfigure and Autoexecute Files on page 1718.

Profile Types

The ExtremeXOS software supports two types of profiles: static and dynamic.

Static Profiles

Static profiles are so named because they are not triggered by dynamic system events. To trigger a static profile, you must enter a command at the switch prompt or run a script that contains the command to start a static profile. The following guidelines apply to static profiles:

- Static profiles are not limited to individual ports and can include system-wide configuration changes.
- Static profiles are not assigned to a port and are not specific to a device or a user.
- Changes made by static profiles are persistent. They are saved in the switch configuration and are preserved during system reboots.

Static profiles are typically used to establish default switch settings. Using scripts and variables, you can create static profiles that serve as templates for initializing switches

or reconfiguring switches to manually respond to network or business events. These templates can simplify complex configuration tasks such as Netlogin.

Dynamic Profiles

Dynamic profiles are so named because they are dynamically triggered by the following types of events:

- Device discovery and disconnect
- User- or standards-based authentication and logoff
- Time of day
- Switch events reported by the EMS

Dynamic profiles are event- or action-driven and do not require an administrator to start the profile.

Without dynamic profile support, IT personnel must be available when devices are added, moved, or changed so they can configure both the network port and the new device. These tasks typically take a long time, do not support mobility, and are often prone to human error.

When dynamic profiles are configured properly and a device connects to an edge port, a triggering event triggers a profile that runs a script to configure the port appropriately. The script can use system run-time variables and information gathered from tools such as Netlogin and *LLDP (Link Layer Discovery Protocol)* to customize the port configuration for the current network environment. For example, the profile can customize the port configuration based on the user ID or MAC address. Dynamic profiles allow you to automate the network response to a variety of network events.

Dynamic profiles create temporary states. For example, if a power outage causes the switch to restart, all ports return to the default configuration. When a triggering event such as a specific device connection occurs again, the profile is applied again. When the device is no longer connected, the disconnect event can trigger another profile to unconfigure the port.

The temporary state configured by a dynamic profile is configured by prepending the `configure cli mode non-persistent` command to the script. The temporary nature of profile configuration is critical for network security. Imagine a situation where a dynamic security profile is used. If the information granting access to specific network resources is saved in the configuration, the switch is restarted, and a user loses network connectivity on a secure port, the secure port still provides network access after the switch restarts. Anybody else can access network resources simply by connecting to that secure port.

Although the switch configuration returns to the default values after a restart, there is no automatic configuration rollback for dynamic profiles. For example, if a profile grants secure access to network resources at user login, the configuration is not automatically rolled back when the user logs off. To roll back the configuration at user log off, you must create another profile that responds to user log off events.

To support configuration rollback, the scripting feature allows you to save information used in dynamic profiles in variables. When a profile is activated and you want the option to roll back to the previous default setting, some information must be saved, such as the default *VLAN (Virtual LAN)* setting or the default configuration of a port. Essentially anything modified from the previous setting can be preserved for future use by the profile that rolls back the configuration.

There can be multiple profiles on a switch, but only one profile runs at a time. Data from a trigger event is used to select the appropriate profile, and that data can also be used to make decision points within a profile. A typical example is the use of a *RADIUS (Remote Authentication Dial In User Service)* server to specify a particular profile and then apply port-based policies to the user based on the user's location.

There is no profile hierarchy and no software validation to detect if a new profile conflicts with older profile. If two profiles conflict, the same profile might produce different results, depending on the events leading up to the profile trigger. When you create profiles, you must be familiar with all profiles on the switch and avoid creating profiles that conflict with each other.

Dynamic Profile Trigger Types

The following sections describe the types of dynamic profile trigger types: device, user authentication, time, and event-management system.

Device Triggers

Device triggers launch a profile when a device connects to or disconnects from a port.

The two types of device triggers are labeled device-detect and device-undetect in the software. Profiles that respond to these triggers are called device-detect profiles or device-undetect profiles.

Typically, a device-detect profile is used to configure a port for the device that has just connected.

Likewise, a device-undetect profile is used to return the port to a default configuration after a device disconnects. A variety of different devices can be connected to a port. When devices connect to the network, Universal Port helps provide the right configuration at the port.

Device triggers respond to the discovery protocols IEEE 802.1ab *LLDP* and ANSI/TIA-1057 LLDP-MED for Voice-over-IP (VoIP) phone extensions. A device-detect trigger occurs when an LLDP packet reaches a port that is assigned to a device-detect profile. A device-undetect trigger occurs when periodically transmitted LLDP packets are not received anymore. LLDP age-out occurs when a device has disconnected or an age-out time has been reached. LLDP must be enabled on ports that are configured for device-detect or device-undetect profiles. LLD P is described in LLDP.

The combination of device triggers and LLDP enables the custom configuration of devices that connect to switch ports. For example, VoIP phones can send and receive information in addition to normal device identification information. The information

sent through LLDP can be used to identify the maximum power draw of the device. The switch can then set the maximum allocated power for that port.

If the switch does not have enough *PoE* left, the switch can take action to lower the PoE loading and try again. The switch can also transmit additional VoIP files and call server configuration information to the phone so the phone can register itself and receive necessary software and configuration information.

There can only be one device-detect profile and one device-undetect profile per port. To distinguish between different connecting devices, you can use if-then-else statements in a profile along with detailed information provided through LLDP.

User Authentication Triggers

User authentication triggers launch a profile when a user or an identified device logs in or out of the network using Network Login on page 1214.

The network login feature does not permit any access beyond the port until the user or device is authenticated.

The two types of user authentication triggers are labeled user-authenticate and user-unauthenticated in the software. Profiles that respond to these triggers are called user-authenticate profiles or user-unauthenticated profiles. Typically, a user-authenticate profile is used to configure a port for a user and device that has just connected. Likewise, a user-unauthenticated profile is used to return the port to a default configuration after a user or device disconnects. Successful network login triggers the user-authenticate profile, and either an explicit logout, a session time out, or a disconnect triggers the user-unauthenticated profile.

Note
VoIP phones are also capable of being authenticated before being allowed on the network. The phone begins 802.1X authentication based on a personal username and password. This authentication step is available and supported by the latest firmware from vendors such as Avaya and Mitel.

Network login requires a *RADIUS* server for user or device authentication.

The RADIUS server provides the following features:

- Centralized database for network authentication
- Further centralization when connected to an LDAP or Active Directory database
- Dynamic switch configuration through Vendor Specific Attributes (VSAs)

VSAs are values that are passed from the RADIUS server to the switch after successful authentication. VSAs can be used by the switch to configure connection attributes such as security policy, *VLAN*, and location. For more information on RADIUS and VSAs, see Security on page 1628.

The following sections introduce each of the network login event types that can trigger profiles:

- 802.1X Network Login

- MAC-Based Network Login
- Web-Based Network Login

802.1X Network Login

Network login 802.1X requires 802.1X client software on the device to be authenticated.

At login, the user supplies a user name and password, which the switch passes to the RADIUS server for authentication. When the user passes authentication, the RADIUS server notifies the switch, and the user-authenticate profile is triggered.

One advantage of 802.1X network login is that it can uniquely identify a user. A disadvantage is that not all devices support 802.1X authentication. For more information, see Network Login on page 1214.

MAC-Based Network Login

MAC-based network login requires no additional software, and it does not require any interaction with the user.

When network login detects a device with a MAC address that is configured on the switch, the switch passes the MAC address and an optional password to the RADIUS server for authentication. When the device passes authentication, the RADIUS server notifies the switch, and the user-authenticate profile is triggered.

One advantage of MAC-based network login is that it requires no special software. A disadvantage is that security is based on the MAC address of the client, so the network is more vulnerable to spoofing attacks. For more information, see Network Login on page 1214.

Note
MAC-based authentication can also be used to identify devices. For example, an entire MAC address or some bits of the MAC address can identify a device and trigger switch port auto-configuration similar to the *LLDP*-based device detect event. The difference between MAC-based authentication and LLDP authentication is that MAC-based authentication does not provide information on the connected device. The advantage of MAC-based authentication is that it enables non-LLDP devices to trigger profiles.

Note
UPM VSA is supported in ONEPolicy in *NetLogin* mode.

Web-Based Network Login

Web-based network login requires a *DHCP (Dynamic Host Configuration Protocol)* server and may require a DNS server.

At login, the user supplies a user name and password through a web browser client, which the switch passes to the RADIUS server for authentication. When the user passes authentication, the RADIUS server notifies the switch, and the user-authenticate profile is triggered.

Some advantages of web-based network login are that it can uniquely identify a user and it uses commonly available web client software. Some disadvantages are a lower level of security and the IP configuration requirement. For more information, see Network Login on page 1214.

Time Triggers

Time triggers launch a profile at a specific time of day or after a specified period of time.

For example, you can use time triggers to launch profiles at the following times:

- 6:00 p.m. every day
- One-time after 15 minutes
- One hour intervals

A profile that uses a time trigger is called a time-of-day profile. You might use a time trigger to launch a profile to disable guest *VLAN* access, shut down a wireless service, or power down a port after business hours. Time triggers enable profiles to perform timed backups for configurations, policies, statistics, and so forth. Anything that needs to happen on a regular basis or at a specific time can be incorporated into a time-of-day profile. Time-of-day profiles are not limited to non-persistent-capable CLI commands and can use any command in the ExtremeXOS CLI.

Unlike the device-detect and user-authenticate triggers, time triggers do not have an equivalent function to the device-undetect or user-unauthenticated triggers. If you need the ability to unconfigure changes made in a time-of-day profile, just create another time-of-day profile to make those changes.

Event Management System Triggers

EMS-event triggers launch a profile when the EMS produces a message that conforms to a predefined definition that is configured on the switch. The ExtremeXOS EMS feature is described in CLEAR-Flow.

Profiles that respond to EMS-event triggers are called EMS-event profiles. Typically, an EMS-event profile is used to change the switch configuration in response to a switch or network event.

The EMS events that trigger Universal Port profiles are defined in EMS filters and can be specified in more detail with additional CLI commands.

You can create EMS filters that specify events as follows:

- Component.subcomponent
- Component.condition
- Component.subcomponent.condition

You can use the `show log components` command to display all the components and subcomponents for which you can filter events. If you specify a filter to take action on a component or subcomponent, any event related to that component triggers the profile. You can use the show log events all command to display all the conditions or

events for which you can filter events. If you decide that you want to configure a profile to take action on an *ACL (Access Control List)* policy change, you can add a filter for the ACL.Policy.Change event.

You can further define an event that triggers a Universal Port profile by specifying an event severity level and text that must be present in an event message.

When a specified event occurs, event information is passed to the Universal Port profile in the form of variables, which can be used to modify the switch configuration.

EMS-triggered profiles allow you to configure responses for any EMS event listed in the `show log components` and `show log events all` commands. However, you must be careful to select the correct event and corresponding response for each profile. For example, if you attempt to create a Universal Port log target for a specific event (component.subcomponent.condition) and you accidentally specify a component (component), the profile is applied to all events related to that component. Using EMS-triggered profiles is similar to switch programming. They provide more control and therefore more opportunity for misconfiguration.

Unlike the device-detect and user-authenticate triggers, EMS-event triggers do not have an equivalent function to the device-undetect or user-unauthenticated triggers.

If you need the ability to unconfigure changes made in an EMS-event profile, just create another static or dynamic profile to make those changes.

How Device-detect Profiles Work

Device-detect profiles enable dynamic port configuration without the use of a *RADIUS* server. Device-detect profiles and device-undetect profiles are triggered as described earlier in Device Triggers on page 2138.

When a device connects to a port that has a device-detect profile configured, the switch runs the specified profile and stops. Only one device detect profile can be configured for a port, so the same profile runs each time a device is detected on the port. Only one device-undetect profile can be configured for a port, so the same profile is run each time the switch detects that all previously-connected devices are no longer connected.

How User Authentication Profiles Work

User-authentication profiles can be assigned to user groups or individual users. Typically, a company creates profiles for groups such as software engineering, hardware engineering, marketing, sales, technical support, operations, and executive. These kinds of categories make profile management more streamlined and simple.

The authentication process starts when a switch receives an authentication request through Network Login. The authentication request can be for a specific user or a MAC address. A user name and password might be entered directly or by means of other security instruments, such as a smart card. A MAC address would be provided by *LLDP*, which would need to be operating on the ingress port. Netogin enforces authentication before granting access to the network. All packets sent by a client on the port do not

go beyond the port into the network until the user is authenticated through a *RADIUS* server.

Starting with ExtremeXOS 30.4, the authentication VLAN information from ONEPolicy is passed to Netlogin, so that UPM scripts can make use of this information. Note that dynamic VLANS are not fully supported, because a dynamic VLAN can be created prior to the client being authenticated. In such cases, the VLAN appears as "<not found> xxx" where "xxx" is the VLAN number, rather than the VLAN name, which usually appears. UPM VLAN processing also does not function, because the VLAN has not yet been created.

The switch authenticates the user through a RADIUS server, which acts as a centralized authorization point for all network devices. The RADIUS server can contain the authentication database, or it can serve as a proxy for a directory service database, such as LDAP or Active Directory. The switch also supports optional backup authentication through the local switch database when a RADIUS server is unavailable.

The RADIUS server responds to the switch and either accepts or denies user authentication. When user authentication is accepted, the RADIUS server can also send Vendor Specific Attributes (VSAs) in the response. The VSAs can specify configuration data for the user such as the Universal Port profile to run for logon, a *VLAN* name, a user location, and a Universal Port profile to run for logout. Extreme Networks has defined vendor specific attributes that specify configuration settings and can include variables to be processed by the Universal Port profile. If profile information is not provided by the RADIUS server (specifically VSA 212 with the UPM profile name), no UPM profile is executed automatically. The UPM user-authentication profile only executes when the RADIUS server returns VSA 212 containing the profile name.

Note
In addition to configuring the UPM user-authentication event, the RADIUS server must return VSA 212 with the UPM profile name. For configuration details, see KB article 000075132.

Profiles are stored and processed on the switch. When a user name or MAC address is authenticated, the switch places the appropriate port in forwarding mode and runs either a profile specified by the RADIUS server, or the profile defined for the authentication event. The profile configures the switch resources for the user and stops running until it is activated again.

When a user or MAC address is no longer active on the network, due to logoff, disconnect, or inactivity, user unauthentication begins.

To complete unauthentication, the switch stops forwarding on the appropriate port and does one of the following:

1. Runs an unauthenticate profile specified by the RADIUS server during authentication.
2. Runs an unauthenticate profile configured on the switch and assigned to the affected port.
3. Runs the authenticate profile initially used to authenticate the user .

The preferred unauthenticate profile is one specified by the RADIUS server during authentication. If no unauthenticate profiles are specified, the switch runs the authenticate profile used to authenticate the user or device.

Profile Configuration Guidelines

You can configure both static and dynamic profiles using the CLI.

Obtaining Profiles

You can write your own profiles.

You can obtain profiles from the Extreme Networks website, another Extreme Networks user or partner, or Extreme Networks professional services.

Sample profiles are listed in Sample Universal Port Configurations on page 2159. The Universal Port Handset Provisioning Module is a collection of profiles and documentation that is available with other samples on the Extreme Networks website.

Profile Rules

All profiles have the following restrictions:

- Maximum 5000 characters in a profile.
- Maximum 128 profiles on a switch.
- Profiles are stored as part of the switch configuration file.
- Copy and paste is the only method to transfer profile data using the CLI.
- Unless explicitly preceded with the command `configure cli mode persistent`, all non-persistent-capable commands operate in non-persistent mode when operating in dynamic profiles.
- Unless explicitly preceded with the command `configure cli mode non-persistent`, all non-persistent-capable commands operate in persistent mode when operating in static profiles.

Note

There is no profile hierarchy, which means users must verify there are no conflicting rules in static and dynamic profiles. This is a normal requirement for ACLs, and is standard when using policy files or dynamic ACLs.

When the switch is configured to allow non-persistent-capable commands to operate in non-persistent mode, the switch configuration related to the non-persistent commands is not saved to the configuration file. This behavior enables the ports to return to their initial state when a reboot or power cycle occurs. You can view the configuration changes made by non-persistent commands through the application's show commands, but the changes do not appear in `show configuration {module-name} {detail}` .

Multiple Profiles on the Same Port

Multiple Universal Port profiles can be created on a switch, but only one profile per event can be applied per port. Different profiles on the same port apply to different events—different authentication events for different devices or users.

You can configure multiple user profiles on a port or a group of ports. For instance, you might create user-authentication profiles for different groups of users, such as Engineering, Marketing, and Sales.

You can also configure a device-triggered profile on a port that supports one or more user profiles. However, you can configure only one device-triggered profile on a port.

Supported Configuration Commands and Functions

Static and dynamic profiles support the full ExtremeXOS command set. They also support the built-in functions described in Using CLI Scripting on page 2182.

Commands that are executed in persistent mode become part of the saved switch configuration that persists when the switch is rebooted. Commands that are executed in non-persistent mode configure temporary changes that are not saved in the switch configuration and do not persist when the switch is rebooted.

However, a subset of these commands operates by default in non-persistent mode when executed in a dynamic profile.

Most commands operate only in persistent mode. The subset of commands that operate in non-persistent mode are called non-persistent-capable commands. The Universal Port feature uses the non-persistent-capable commands to configure temporary changes that could create security issues if the switch were rebooted or reset. The use of non-persistent-capable commands in scripts and Universal Port profiles allows you to make temporary configuration changes without affecting the default configuration the next time the switch is started.

The following table shows the non-persistent capable CLI commands.

Non-Persistent-Capable Configuration Commands

ACL Commands

```
configure access-list add dynamic_rule [ [[first | last] {priority
p_number} {zone zone} ] | [[before | after] rule] | [ priority p_number
{zone zone} ]] [ any | vlan vlanname | ports portlist ] {ingress |
egress}
```

```
configure access-list delete ruleName [ any | vlan vlanname | ports
portlist | all] {ingress | egress}
```

LLDP Commands

```
configure lldp ports portlist [advertise | dont advertise | no-advertise
| dcbx] {all-tlvs | management-address | port-description | system-
capabilities | system-description | system-name | vendor-specific}
```

Port Commands

```
disable port [port_list | all]

disable jumbo-frame ports [all | port_list]

enable port [port_list | all]

enable jumbo-frame ports [all | port_list]
```

Power over Ethernet Commands

```
configure inline-power label string ports port_list

configure inline-power operator-limit milliwatts ports [all |port_list]

configure inline-power priority [critical | high | low] ports port_list

disable inline-power

disable inline-power ports [all | port_list]

disable inline-power slot slot

enable inline-power

enable inline-power ports [all | port_list]

enable inline-power slot slot

unconfigure inline-power priority ports [all | port_list]
```

VLAN Commands

```
configure {vlan} vlan_name add ports [port_list | all] {tagged |
untagged} {{stpd} stpd_name} {dot1d | emistp | pvst-plus}}

configure ip-mtu mtu vlan vlan_name
```

QOS/Rate-limiting Commands

```
configure ports port_list {qosprofile} qosprofile
```

Show Commands

All show commands can be executed in non-persistent mode.

By default, all commands operate in persistent mode with the following exceptions:

- In Universal Port dynamic profiles, the non-persistent-capable commands operate in non-persistent mode unless preceded by the `configure cli mode persistent` command in the profile.
- In the CLI, CLI scripts, and static profiles, the non-persistent-capable commands operate in non-persistent mode only when preceded by the `configure cli mode non-persistent` command.

You can use the `configure cli mode persistent` command and the `configure cli mode non-persistent` command to change the mode of operation for non-persistent-capable commands multiple times within a script, profile, or configuration session.

Universal Port Variables

Universal Port uses CLI Scripting variables to make system and trigger event information available to profiles. For more information, see Using CLI Scripting on page 2182.

Variables allow you to create profiles and scripts that respond to the state of the switch as defined in the variables. When a profile is triggered, the system passes variables to the profile. You can also create and use variables of your own. User-defined variables are limited to the current context unless explicitly saved.

Note
You must enable CLI scripting before using variables or executing a script.

If you save variables (as described in Saving, Retrieving, and Deleting Session Variables on page 2193), certain data from one profile can be reused in another profile for another event. For example, between login and logout events, the data necessary for the rollback of a port configuration can be shared.

The following sections describe the variables that are available to profiles:

- Common Variables
- User Profile Variables
- Device Detect Profile Variables
- Event Profile Variables

Common Variables

Table 166 shows the variables that are always available for use by any script. These variables are set up for use before a script or profile is executed.

Table 166: Common Variables

Variable Syntax	Definition
$STATUS	Status of last command execution.
$CLI.USER	Username for the user who is executing this CLI.
$CLI.SESSION_ID	An identifier for a session. This identifier is available for the roll-back event when a device or user times out.
$CLI.SESSION_TYPE	Type of session of the user.
$EVENT.NAME	This is the event that triggered this profile.
$EVENT.TIME	Time this event occurred. The time is in seconds since epoch.

Table 166: Common Variables (continued)

Variable Syntax	Definition
$EVENT.TIMER_TYPE	Type of timer, which is periodic or non_periodic.
$EVENT.TIMER_NAME	Name of the timer that the Universal Port is invoking.
$EVENT.TIMER_LATE_SECS	Time difference between when the timer fired and when the actual shell was run in seconds.
$EVENT.PROFILE	Name of the profile that is being run currently.

User Profile Variables

Table 167 shows the variables available to user profiles.

Table 167: User Profile Variables

Variable Syntax	Definition
$EVENT.USERNAME	Name of user authenticated. This is a string with the MAC address for MAC-based user-login.
$EVENT.NUMUSERS	Number of authenticated supplicants on this port after this event occurred. **Note:** For user-authenticated events, the initial value of this variable is 0. For user unauthenticated events, the initial value is 1.
$EVENT.USER_MAC	MAC address of the user.
$EVENT.USER_PORT	Port associated with this event.
$EVENT.USER_VLAN	*VLAN* associated with this event or user. **Note:** If the VLAN is assigned dynamically, and the name is not available at the time of authentication, "$EVENT.USER_VLAN" is returned as "<Not Found> xxx" where xxx is the VLAN number. This is not translated by UPM VLAN processing and appears for diagnostic purposes only.
$EVENT.USER_ALL_VLANS	When a user is authenticated to multiple VLANs, this variable includes all VLANs for which the user is authenticated.
$EVENT.USER_IP	IP address of the user if applicable. Otherwise, this variable is blank.

Device Detect Profile Variables

Table 168 shows the variables available to device detect profiles.

Table 168: Device Profile Variables

Variable Syntax	Definition
$EVENT.DEVICE	Device identification string. Possible values for EVENT.DEVICE are: AVAYA_PHONE, GEN_TEL_PHONE, ROUTER, BRIDGE, REPEATER, WLAN_ACCESS_PT, DOCSIS_CABLE_SER, STATION_ONLY and OTHER. These strings correspond to the devices that the *LLDP* application recognizes and reports to the Universal Port management application.
$EVENT.DEVICE_IP	The IP address of the device (if available). Blank if not available.
$EVENT.DEVICE_MAC	The MAC address of the device (if available). Blank if not available.
$EVENT.DEVICE_POWER	The power of the device in milliwatts (if available). Blank if not available.
$EVENT.DEVICE_MANUFACTURER_NAME	The manufacturer of the device.
$EVENT.DEVICE_MODEL_NAME	Model name of the device.
$EVENT.USER_PORT	Port associated with the event.

Event Profile Variables

Table 169 shows the variables available to event profiles.

Table 169: Event Profile Variables

Variable Syntax	Definition
$EVENT.NAME	The event message.
$EVENT.LOG_DATE	The event date.
$EVENT.LOG_TIME	The event time.
$EVENT.LOG_COMPONENT_SUBCOMPONENT	The component and subcomponent affected by the event as it appears in the show log components command display.
$EVENT.LOG_EVENT	The event condition as it appears in the show log events command display.
$EVENT.LOG_FILTER_NAME	The EMS filter that triggered the profile.
$EVENT.LOG_SEVERITY	The event severity level defined in EMS.

Table 169: Event Profile Variables (continued)

Variable Syntax	Definition
$EVENT.LOG_MESSAGE	The event message with arguments listed in the format %1%.
$EVENT.LOG_PARAM_0 to $EVENT.LOG_PARAM_9	Event arguments 0 to 9.

Collecting Information from Supplicants

A supplicant is a device such as a VoIP phone or workstation that connects to the switch port and requests network services.

As described in LLDP, LLDP is a protocol that can be used to collect information about device capabilities from attached devices or supplicants.

To use Universal Port with LLDP, you must enable LLDP on the port.

Note

Avaya and Extreme Networks have developed a series of extensions for submission to the standards consortium for inclusion in a later version of the LLDP-MED standard:

- Avaya Power conservation mode
- Avaya file server
- Avaya call server

The following is an example of information provided through LLDP about an IP phone:

```
LLDP Port 1 detected 1 neighbor
Neighbor: (5.1)192.168.10.168/00:04:0D:E9:AF:6B, age 7 seconds
- Chassis ID type: Network address (5); Address type: IPv4 (1)
Chassis ID     : 192.168.10.168
- Port ID type: MAC address (3)
Port ID     : 00:04:0D:E9:AF:6B
- Time To Live: 120 seconds
- System Name: "AVAE9AF6B"
- System Capabilities : "Bridge, Telephone"
Enabled Capabilities: "Bridge, Telephone"
- Management Address Subtype: IPv4 (1)
Management Address        : 192.168.10.168
Interface Number Subtype  : System Port Number (3)
Interface Number          : 1
Object ID String          : "1.3.6.1.4.1.6889.1.69.1.13"
- IEEE802.3 MAC/PHY Configuration/Status
Auto-negotiation       : Supported, Enabled (0x03)
Operational MAU Type   : 100BaseTXFD (16)
- MED Capabilities: "MED Capabilities, Network Policy, Inventory"
MED Device Type : Endpoint Class III (3)
- MED Network Policy
Application Type  : Voice (1)
Policy Flags      : Known Policy, Tagged (0x1)
VLAN ID           : 0
L2 Priority       : 6
DSCP Value        : 46
```

```
- MED Hardware Revision: "4625D01A"
- MED Firmware Revision: "b25d01a2_7.bin"
- MED Software Revision: "a25d01a2_7.bin"
- MED Serial Number: "061622014487"
- MED Manufacturer Name: "Avaya"
- MED Model Name: "4625"
- Avaya/Extreme Conservation Level Support
Current Conservation Level: 0
Typical Power Value       : 7.4 Watts
Maximum Power Value       : 9.8 Watts
Conservation Power Level  : 1=7.4W
- Avaya/Extreme Call Server(s): 192.168.10.204
- Avaya/Extreme IP Phone Address: 192.168.10.168 255.255.255.0
Default Gateway Address       : 192.168.10.254
- Avaya/Extreme CNA Server: 0.0.0.0
- Avaya/Extreme File Server(s): 192.168.10.194
- Avaya/Extreme IEEE 802.1q Framing: Tagged
```

Note
LLDP is tightly integrated with IEEE 802.1X authentication at edge ports. When used together, LLDP information from authenticated end point devices is trustable for automated configuration purposes. This tight integration between 802.1X and LLDP protects the network from automation attacks.

Supplicant Configuration Parameters

As described in LLDP, LLDP is a protocol that can be used to configure attached devices or supplicants. The following LLDP parameters are configurable on the switch ports when device-detect profiles execute are:

- *VLAN* Name
- Port VLAN ID
- Power Conservation Mode
- Avaya File Server
- Avaya Call server
- 802.1Q Framing

Universal Port Configuration Overview

Because Universal Port operates with multiple ExtremeXOS software features and can operate with multiple third-party products, Universal Port configuration can require more than just the creation of profiles and triggers.

No single overview procedure can cover all the possible Universal Port configurations. The following sections provide overviews of the common types of Universal Port configurations.

Device-Detect Configurations

A Universal Port device-detect configuration requires only a switch and supplicants. If *PoE* devices will connect to the switch, the switch should support PoE. Supplicants should support *LLDP* in the applicable software or firmware.

Note
To support supplicant configuration, you might consider adding a *DHCP* server to your network.

Use the following procedure to configure Universal Port for device detection:

1. Create a device-detect profile as described in Creating and Configuring New Profiles.
2. Create a device-undetect profile as described in Creating and Configuring New Profiles.
3. Assign the device-detect profile to the edge ports as described in Configuring a Device Event Trigger.
4. Assign the device-undetect profile to the edge ports as described in Configuring a Device Event Trigger.
5. Verify that correct profiles are assigned to correct ports using the following command:

   ```
   show upm event event-type
   ```
6. Enable LLDP message advertisements on the ports that are configured for device-detect profiles as described in LLDP.
7. Test profile operation as described in Verifying a Universal Port Profile on page 2158.

User-Authentication Configurations

A Universal Port user-authenticate configuration requires specific components:

- An Extreme Networks switch, which might need to include *PoE* support.
- *RADIUS* server for user authentication and VSA transmission.
- Supplicants that support the authentication method you select. *LLDP* support is recommended, but is optional when MAC address authentication is used.

Note
To support supplicant configuration, you might consider adding a *DHCP* server to your network. For VoIP applications, you can use a TFTP server and a call server to provide for additional supplicant configuration.

Use the following procedure to configure Universal Port for user login:

1. Configure the RADIUS server as described in Security on page 1628. The configuration should include the following:
   - User ID and password for RADIUS clients.
   - Extreme Networks custom VSAs.
   - Addition of the edge switch as a RADIUS client.
2. Create a user-authenticate profile as described in Creating and Configuring New Profiles.

3. Create a user-unauthenticate profile as described in Creating and Configuring New Profiles.
4. Assign the user-authenticate profile to the edge ports as described in Configuring a User Login or Logout Event Trigger.
5. Assign the user-unauthenticate profile to the edge ports as described in Configuring a User Login or Logout Event Trigger.
6. Configure network login on the edge switch as described in Network Login.
7. Configure the edge switch as a RADIUS client as described in Security.
8. Verify that correct profiles are assigned to correct ports by entering the following command:

   ```
   show upm event event-type
   ```

9. Enable LLDP message advertisements on the ports that are configured for device-detect profiles as described in LLDP Overview.
10. Test profile operation as described in Verifying a Universal Port Profile on page 2158.

Time-of-Day Configurations

To configure Universal Port to use a time-of-day profile, use the following procedure:

1. Create a profile as described in Creating and Configuring New Profiles.
2. Create and configure a timer as described in Configuring a Universal Port Timer.
3. Create the timer trigger and attach it to the profile as described in Configuring a Timer Trigger.

EMS-Event Configurations

To configure Universal Port to use an EMS-event profile, use the following procedure:

1. Create the EMS-Event profile as described in Creating and Configuring New Profiles.
2. Create and configure an event filter to identify the trigger event as described in Creating an EMS Event Filter.
3. Create the event trigger and attach it to the profile and filter as described in Configuring an EMS Event Trigger.
4. Enable the event trigger as described in Enabling and Disabling an EMS Event Trigger on page 2156.

Using Universal Port in an LDAP or Active Directory Environment

The *RADIUS* server can operate in proxy mode with information stored in a central directory service such as LDAP or Active Directory.

This proxy mode is configured between the RADIUS server and the central directory service. Once configured, supplicants can be authenticated from the central directory service.

For more information, see the following:

- Setting Up Open LDAP on page 1638.
- RADIUS server product documentation

- Product documentation for your central directory service

Configuring Universal Port Profiles and Triggers

You can configure both static and dynamic profiles using the CLI.

Note
In the CLI, “upm” is used as an abbreviation for the Universal Port feature.

Creating and Configuring New Profiles

When you create and configure a new profile, you are basically writing a script within a profile that can be triggered by system events. For more information on the rules, commands, and variables that apply to profiles, see Profile Configuration Guidelines on page 2144.

1. Create and configure a new profile using the following command:
   `configure upm profile profile-name maximum execution-time seconds`
2. After you enter the command, the switch prompts you to add command statements to the profile as shown in the following example:

```
switch # create upm profile detect-voip
Start typing the profile and end with a . as the first and the only character on a
line.
Use - edit upm profile <name> - for block mode capability
create log message Starting_Script_DETECT-voip
set var callServer 192.168.10.204
set var fileServer 192.168.10.194
set var voiceVlan voice
set var CleanupProfile CleanPort
set var sendTraps false
#
.
switch #
```

   The example above creates a log entry and sets some variables, but it is not complete. This example shows that after you enter the `create upm profile` command, you can enter system commands. When you have finished entering commands, you can exit the profile creation mode by typing the period character (.) at the start of a line and pressing **[Enter]**.

Editing an Existing Profile

- Edit an existing profile.
  `edit upm profile profile-name`

Configuring a Device Event Trigger

There are two types of device event triggers, which are named as follows in the CLI: device-detect and device-undetect. When you configure a device event trigger, you assign one of the two device event trigger types to a profile and specify the ports to which the triggered profile applies.

To configure a device event trigger, use the following command:

```
configure upm event upm-event profile profile-name ports port_list
```

Replace *upm-event* with one of the device event trigger types: device-detect or device-undetect.

Configuring a User Login or Logout Event Trigger

There are two types of user event triggers, which are named as follows in the CLI: user-authenticate and user-unauthenticated. When you configure a user event trigger, you assign one of the two user event trigger types to a profile and specify the ports to which the triggered profile applies.

To configure a user event trigger, use the following command:

```
configure upm event upm-event profile profile-name ports port_list
```

Replace *upm-event* with one of the device event trigger types: user-authenticate or user-unauthenticated.

Configuring a Universal Port Timer

To configure a Universal Port timer, you must complete two steps:

1. Create the timer by using the following command:
   ```
   create upm timer timer-name
   ```
2. Configure the timer by using the following commands:
   ```
   configure upm timer timer-name after time-in-secs {every seconds}

   configure upm timer timer-name at month day year hour min secs {every seconds}
   ```

Configuring a Timer Trigger

When you configure a timer trigger, you assign a configured timer to a profile. When the configured time arrives, the switch executes the profile.

To configure a timer trigger, use the following command:

```
configure upm timer timerName profile profileName
```

Replace *timerName* with the timer name and *profileName* with the profile name.

Creating an EMS Event Filter

An EMS event filter identifies an event that can be used to trigger a profile. To create an EMS event filter, use the following procedure:

1. Create a log filter to identify the event by using the following command:
   ```
   create log filter name {copy filter name}
   ```

2. Configure the log filter using the following commands:

   ```
   configure log filter name [add | delete] {exclude} events [event-condition | [all | event-component] {severity severity {only}}]
   ```

   ```
   configure log filter name [add | delete] {exclude} events [event-condition | [all | event-component] {severity severity {only}}] [match | strict-match] type value
   ```

Configuring an EMS Event Trigger

When you configure an EMS event trigger, you identify an EMS filter that defines the event and a profile that runs when the event occurs. To configure an EMS event-triggered profile, use the following procedure:

1. Create a log target to receive the event notification by using the following command:

   ```
   create log target upm {upm_profile_name}
   ```

2. Configure the log target to specify a filter and any additional parameters that define the event by using the following commands:

   ```
   configure log target upm {upm_profile_name} filter filter-name {severity [[severity] {only}]}
   ```

   ```
   configure log target upm {upm_profile_name} match {any | regex}
   ```

Enabling and Disabling an EMS Event Trigger

When you configure an EMS event trigger, it is disabled.

To enable an EMS event trigger or disable a previously enabled trigger, use the following commands:

```
enable log target upm {upm_profile_name}
```

```
disable log target upm {upm_profile_name}
```

Unconfiguring a Timer

To unconfigure a timer, use the following command:

```
unconfigure upm timer timerName profile profileName
```

Managing Profiles and Triggers

Use the following actions to manage profiles and triggers.

Manually Executing a Static or Dynamic Profile

Profiles can be run from the CLI by configuring the system to run as it would when the trigger events happen. This facility is provided to allow you to test how the system

behaves when the actual events happen. The actual configuration is applied to the switch when the profile is run.

Note

Variables are not validated for correct syntax.

To manually execute a profile, use the following command:

```
run upm profile profile-name {event event-name} {variables variable-string}
```

Example:

```
run upm profile afterhours
```

If the variables keyword is not present, but an event variable is specified, you are prompted for various environment variables appropriate for the event, including the VSA string for user authentication.

Displaying a Profile

To display a profile, use the following command:

```
show upm profile name
```

Displaying Timers

Display a list of timers and associated timer information.

```
show upm timers
```

Displaying Universal Port Events

Display events for trigger types. You can display a list of events that relate to one of the following trigger types:

- device-detect
- device-undetect
- user-authenticate
- user-unauthenticated

To display a list of Universal Port events for one of the above triggers, use the following command:

```
show upm event event-type
```

Replace *event-type* with one of the trigger types listed above.

Displaying Profile History

To display a list of triggered events and associate event data, enter one of the following commands:

```
show upm history {profile profile-name | event upm-event | status [pass
| fail] | timer timer-name | detail}

show upm history exec-id number
```

Verifying a Universal Port Profile

To verify a Universal Port profile configuration, trigger the profile and verify that it works properly. Trigger the profile based on the trigger type as follows:

- Device triggers–plug in the device
- Authentication triggers–authenticate a device or user
- Timer triggers–temporarily configure the timer for an approaching time
- EMS event triggers–reproduce the event to which the trigger responds

You can use the commands described earlier in this section to view information about the profile and how it behaves.

Because Universal Port works with multiple switch features, you might want to enter commands to examine the configuration of those features.

The following commands are an example of some of the commands that can provide additional information about profile operation.

Run:

```
show lldp
show lldp neighbors
show log
show netlogin
```

Handling Profile Execution Errors

To conserve resources, the switch stores only the last execution log for the profile that resulted in an error.

The status column displays the status of the last command executed in a profile. If the last command was successful, the status shows “pass”; otherwise, the status shows “failed”. If you wish to terminate the execution of a profile on encountering the first error, you must include the line “`configure cli scripting mode abort-on-error`” in the beginning of the profile. Otherwise, the profile executes all the commands even if an error was encountered and displays the status of the last command executed.

- To see a tabular display showing the complete history of the last 100 profiles, use the following command:

  ```
  show upm history {profile profile-name | event upm-event | status
  [pass | fail] | timer timer-name | detail}
  ```

Use the detail keyword to display the actual executions that happened when the profile was run.

- To display a specific execution that was run, use the following command:

  ```
  show upm history exec-id number
  ```

  Select the exec-id number from the list in the tabular display.

Disabling and Enabling a Profile

Universal Port profiles are automatically enabled when they are created.

To disable a profile or enable a previously disabled profile, use the following commands:

```
disable upm profile profile-name
```

```
enable upm profile profile-name
```

Deleting a Profile

To delete a profile, use the following command:

```
delete upm profile profile-name
```

Deleting a Timer

To delete a timer, use the following command:

```
delete upm timer timer-name
```

Deleting an EMS Event Trigger

To delete an EMS event trigger, use the following command:

```
delete log target upm {upm_profile_name}
```

Sample Universal Port Configurations

The following sections provide example configurations for a switch, profile, and policy file.

Sample MAC Tracking Profile

The example in this section shows how to create a profile that takes action based on the MAC tracking feature. When the MAC tracking feature detects a MAC move in a *VLAN*, the MAC tracking feature generates an EMS log, which then triggers a profile.

Note

You can also use the Identity Management feature to configure ports in response to MAC device detection events. For more information, see Identity Management.

Switch Configuration

The general switch configuration is as follows:

```
#Vlan  config
create vlan v1
configure v1 add ports 1:17-1:18
configure vlan v1 ipadd 192.168.10.1/24
#mac tracking config
create fdb mac-tracking entry 00:01:02:03:04:01
create fdb mac-tracking entry 00:01:02:03:04:02
create fdb mac-tracking entry 00:01:02:03:04:03
create fdb mac-tracking entry 00:01:02:03:04:04
create fdb mac-tracking entry 00:01:02:03:04:05
#Log filter configuration
create log filter macMoveFilter
configure log filter "macMoveFilter" add events "FDB.MACTracking.MACMove"
#Meter configuration for ingress /egress rate limit
create meter m1
configure meter m1 peak-rate 250 mbps
create meter m2
configure meter m2 peak-rate 500 mbps
```

MAC Tracking EMS Log Message

The MAC tracking feature produces the following EMS log message and message parameters:

```
The MAC address %0% on VLAN '%1%' has moved from port %2% to port %3%"
EVENT.LOG_PARAM_1 "vlan name"
EVENT.LOG_PARAM_2 "source port"
EVENT.LOG_PARAM_3 "moved port"
```

Profile Configuration

The profile is configured as follows:

```
create upm profile macMove   ;# editor
enable cli scripting
create access-list dacl1 "source-address  192.168.10.0/24 " "permit ;count dacl1"
create access-list dacl2 "source-address  192.168.11.0/24 " "permit ;count dacl2"
create access-list dacl3 "source-address  192.168.15.0/24 " "deny ;count dacl3"
create access-list dacl4 "source-address  192.168.16.0/24 " "deny ;count dacl4"
create access-list dacl5 "source-address  192.168.17.0/24 " "deny ;count dacl5"
configure access-list add dacl1 first ports $(EVENT.LOG_PARAM_3)
configure access-list add dacl2 first ports $(EVENT.LOG_PARAM_3)
```

```
configure access-list add dacl3 first ports $(EVENT.LOG_PARAM_3)
configure access-list add dacl4 first ports $(EVENT.LOG_PARAM_3)
configure access-list add dacl5 first ports $(EVENT.LOG_PARAM_3)
conf access-list ingress_limit vlan v1
conf access-list ingress_limit ports $(EVENT.LOG_PARAM_3)
conf access-list egress_limit any   ;# enter . for SAVE/EXIT
log target configuration
create log target upm "macMove"
configure log target upm "macMove" filter "macMoveFilter"
enable log target upm "macMove"
```

Policy File Configuration

This example uses the following two policy files:

```
Ingress rate limit (ingress_limit.pol)
================================
entry ingress {
if {
ethernet-source-address 00:AA:00:00:00:01;
ethernet-destination-address 00:BB:00:00:00:01;
} then {
Meter m1;
count c1;
}
}
Egress QoS (egress_limit.pol)
================================
entry egress {
if {
ethernet-source-address 00:BB:00:00:00:01;
ethernet-destination-address 00:AA:00:00:00:01;
} then {
qosprofile qp2;
count c2;
}
}
```

Console Logs

The following show commands display the switch configuration:

```
* (debug) BD-12804.7 # show log con fil
Log Filter Name: DefaultFilter
I/                                                  Severity
E Component   SubComponent Condition                CEWNISVD
- ----------- ------------ ----------------------- --------
I All                                               ********
Log Filter Name: macMoveFilter
I/                                                  Severity
E Component   SubComponent Condition                CEWNISVD
- ----------- ------------ ----------------------- --------
I FDB         MACTracking  MACMove                 ---N----
* (debug) BD-12804.14 # sh fdb mac-tracking configuration
SNMP trap notification    : Disabled
MAC address tracking table (5 entries):
00:01:02:03:04:01
00:01:02:03:04:02
00:01:02:03:04:03
00:01:02:03:04:04
00:01:02:03:04:05
* (debug) BD-12804.15 #
```

```
* (debug) BD-12804.27 # show meter
-------------------------------------------------------------------------------
Name   Committed Rate(Kbps) Peak Rate(Kbps)   Burst Size(Kb)
-------------------------------------------------------------------------------
m1                     --         250000          --
m2                     --         500000          --
Total number of Meter(s) : 2
* (debug) BD-12804.28 #
```

The following show commands display the switch status after a MAC address move:

```
==================================
(debug) BD-12804.7 # show log
05/14/2009 11:33:54.89 <Noti:ACL.Policy.bind> MSM-A:
Policy:bind:egress_limit:vlan:*:port:*:
05/14/2009 11:33:54.89 <Info:pm.config.loaded> MSM-A: Loaded Policy: egress_limit number
of entries 1
05/14/2009 11:33:54.89 <Info:pm.config.openingFile> MSM-A: Loading policy egress_limit
from file /config/egress_limit.pol
05/14/2009 11:33:54.89 <Noti:ACL.Policy.bind> MSM-A:
Policy:bind:ingress_limit:vlan:*:port:1:18:
05/14/2009 11:33:54.88 <Noti:ACL.Policy.bind> MSM-A:
Policy:bind:ingress_limit:vlan:v1:port:*:
05/14/2009 11:33:54.87 <Info:pm.config.loaded> MSM-A: Loaded Policy: ingress_limit number
of entries 1
05/14/2009 11:33:54.87 <Info:pm.config.openingFile> MSM-A: Loading policy ingress_limit
from file /config/ingress_limit.pol
05/14/2009 11:33:54.72 <Noti:UPM.Msg.upmMsgExshLaunch> MSM-A: Launched profile macMove
for the event log-message
A total of 8 log messages were displayed.
* (debug) BD-12804.8 # show upm history
-------------------------------------------------------------------------------
Exec   Event/              Profile          Port Status Time Launched
Id     Timer/ Log filter
-------------------------------------------------------------------------------
1      Log-Message(macMoveF macMove          --- Pass    2009-05-14 11:33:54
-------------------------------------------------------------------------------
Number of UPM Events in Queue for execution: 0
* (debug) BD-12804.9 # sh upm history detail
UPM Profile: macMove
Event:  Log-Message(macMoveFilter)
Profile Execution start time: 2009-05-14 11:33:54
Profile Execution Finish time: 2009-05-14 11:33:54
Execution Identifier: 1 Execution Status: Pass
Execution Information:
1 # enable cli scripting
2 # configure cli mode non-persistent
3 # set var EVENT.NAME LOG_MESSAGE
4 # set var EVENT.LOG_FILTER_NAME "macMoveFilter"
5 # set var EVENT.LOG_DATE "05/14/2009"
6 # set var EVENT.LOG_TIME "11:33:54.72"
7 # set var EVENT.LOG_COMPONENT_SUBCOMPONENT "FDB.MACTracking"
8 # set var EVENT.LOG_EVENT "MACMove"
9 # set var EVENT.LOG_SEVERITY "Notice"
10 # set var EVENT.LOG_MESSAGE "The MAC address %0% on VLAN '%1%' has moved from port %2%
to port %3%"
11 # set var EVENT.LOG_PARAM_0 "00:01:02:03:04:05"
12 # set var EVENT.LOG_PARAM_1 "v1"
13 # set var EVENT.LOG_PARAM_2 "1:17"
14 # set var EVENT.LOG_PARAM_3 "1:18"
15 # set var EVENT.PROFILE macMove
16 # enable cli scripting
17 # create access-list dacl1 "source-address  192.168.10.0/24 " "permit ;count dacl1"
```

```
18 # create access-list dacl2 "source-address  192.168.11.0/24 " "permit ;count dacl2"
19 # create access-list dacl3 "source-address  192.168.15.0/24 " "deny ;count dacl3"
20 # create access-list dacl4 "source-address  192.168.16.0/24 " "deny ;count dacl4"
21 # create access-list dacl5 "source-address  192.168.17.0/24 " "deny ;count dacl5"
22 # configure access-list add dacl1 first ports $(EVENT.LOG_PARAM_3)
done!
23 # configure access-list add dacl2 first ports $(EVENT.LOG_PARAM_3)
done!
24 # configure access-list add dacl3 first ports $(EVENT.LOG_PARAM_3)
done!
25 # configure access-list add dacl4 first ports $(EVENT.LOG_PARAM_3)
done!
26 # configure access-list add dacl5 first ports $(EVENT.LOG_PARAM_3)
done!
27 # conf access-list ingress_limit vlan v1
done!
28 # conf access-list ingress_limit ports $(EVENT.LOG_PARAM_3)
done!
29 # conf access-list egress_limit any
done!
--------------------------------------------------------------------------------
Number of UPM Events in Queue for execution: 0
* (debug) BD-12804.10 #
* (debug) BD-12804.7 # show fdb mac-tracking statistics
MAC Tracking Statistics      Thu May 14 11:41:10 2009
Add         Move        Delete
MAC Address        events      events      events
=====================================================
00:01:02:03:04:01           0           0           0
00:01:02:03:04:02           0           0           0
00:01:02:03:04:03           0           0           0
00:01:02:03:04:04           0           0           0
00:01:02:03:04:05           1           1           0
=====================================================
0->Clear Counters  U->page up  D->page down ESC->exit
(debug) BD-12804.5 # show access-list
Vlan Name    Port   Policy Name          Dir      Rules  Dyn Rules
=====================================================================
*            *      egress_limit         ingress  1      0
*            1:18   ingress_limit        ingress  1      5
v1           *      ingress_limit        ingress  1      0
* (debug) BD-12804.6 # show access-list dynamic
Dynamic Rules: ((*)- Rule is non-permanent )
(*)dacl1                          Bound to 1 interfaces for application Cli
(*)dacl2                          Bound to 1 interfaces for application Cli
(*)dacl3                          Bound to 1 interfaces for application Cli
(*)dacl4                          Bound to 1 interfaces for application Cli
(*)dacl5                          Bound to 1 interfaces for application Cli
(*)hclag_arp_0_4_96_1e_32_80      Bound to 0 interfaces for application HealthCheckLAG
* (debug) BD-12804.7 #
* (debug) BD-12804.7 #
================================================================================
```

Universal Port Handset Provisioning Module Profiles

The Universal Port Handset Provisioning Module provides the following profiles:

- Device-Triggered Generic Profile on page 2164
- Authentication-Triggered Avaya Profile on page 2166

Device-Triggered Generic Profile

This is a template for configuring network parameters for VoIP phone support without 802.1X authentication. The profile is triggered after an *LLDP* packet is detected on the port.

```
#*******************************
# Last Updated: April 11, 2007
# Tested Phones: Avaya 4610, 4620, 4625
# Requirements: LLDP capable devices
#*******************************
# @MetaDataStart
# @ScriptDescription "This is a template for configuring network parameters for VoIP
phones support LLDP but without authentication.  The module is triggered through the
detection of an LLDP packet on the port.   The following network side configuration is
done: enable SNMP traps, QOS assignment, adjust POE reservation values based on device
requirements, add the voiceVlan to the port as tagged. "
# @VariableFieldLabel "Voice VLAN name"
set var voicevlan voiceavaya
# @VariableFieldLabel "Send trap when LLDP event happens (true or false)"
set var sendTraps false
# @VariableFieldLabel "Set QoS Profile (true or false)"
set var setQuality false
# @MetaDataEnd
#
if (!$match($EVENT.NAME,DEVICE-DETECT)) then
create log message Starting_LLDP_Generic_Module_Config
# VoiceVLAN configuration
configure vlan $voicevlan add port $EVENT.USER_PORT tagged
#SNMP Trap
if (!$match($sendTraps,true)) then
create log message Config_SNMP_Traps
enable snmp traps lldp ports $EVENT.USER_PORT
enable snmp traps lldp-med ports $EVENT.USER_PORT
else
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
#Link Layer Discovery Protocol-Media Endpoint Discover
create log message Config_LLDP
configure lldp port $EVENT.USER_PORT advertise vendor-specific med capabilities
configure lldp port $EVENT.USER_PORT advertise vendor-specific dot1 vlan-name vlan
$voicevlan
configure lldp port $EVENT.USER_PORT advertise vendor-specific med policy application
voice vlan $voicevlan dscp 46
configure lldp port $EVENT.USER_PORT advertise vendor-specific med power-via-mdi
#Configure POE settings per device requirements
create log message Config_POE
configure inline-power operator-limit $EVENT.DEVICE_POWER ports $EVENT.USER_PORT
#QoS Profile
if (!$match($setQuality,true)) then
create log message Config_QOS
configure port $EVENT.USER_PORT qosprofile qp7
endif
endif
if (!$match($EVENT.NAME,DEVICE-UNDETECT) && $match($EVENT.DEVICE_IP,0.0.0.0)) then
create log message Starting_LLDP_Generic_UNATUH_Module_Config
if (!$match($sendTraps,true)) then
create log message UNConfig_SNMP_Traps
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
create log message UNConfig_LLDP
```

```
unconfig lldp port $EVENT.USER_PORT
if (!$match($setQuality,true)) then
create log message UNConfig_QOS
unconfig qosprofile ports $EVENT.USER_PORT
endif
unconfig inline-power operator-limit ports $EVENT.USER_PORT
endif
if (!$match($EVENT.NAME,DEVICE-UNDETECT) && !$match($EVENT.DEVICE_IP,0.0.0.0)) then
create log message DoNothing_0.0.0.0
create log message $EVENT.TIME
endif
create log message End_LLDP_Generic_Module_Config
```

Authentication-Triggered Generic Profile

This profile has been created for phones that support an authentication protocol and assumes that the phone does not support *LLDP* and is provisioned using *DHCP* options.

This is a template for configuring network parameters for 802.1X authenticated devices. The module is triggered through successful authentication or unauthentication of the device.

```
#*******************************************
# Last Updated: April 11, 2007
# Tested Phones: Avaya 4610, 4620, 4625
# Requirements: 802.1X capable devices, netlogin configured and enabled on deployment
ports
#*******************************************
# @MetaDataStart
# @ScriptDescription "This is a template for configuring network parameters for 802.1X
authenticated devices.  The module is triggered through successful authentication of the
device.  The following network side configuration is done: QOS assignment and enables DOS
protection.  When used with IP phones, phone provisioning is done through DHCP options."
# @Description "VLAN name to add to port"
set var vlan1 voiceavaya
# @VariableFieldLabel "Set QoS Profile (yes or no)"
set var setQuality yes
# @Description "QoS Profile (0-100)"
set var lowbw 50
# @VariableFieldLabel "QoS MAX Bandwidth (0-100)"
set var highbw 100
# @VariableFieldLabel "Enable Denial of Service Protection (yes or no)"
set var dosprotection yes
# @MetaDataEnd
##################################
# Start of USER-AUTHENTICATE block
##################################
if (!$match($EVENT.NAME,USER-AUTHENTICATED)) then
#############
#QoS Profile
#############
# Adds a QOS profile to the port
if (!$match($setQuality,yes)) then
create log message Config_QOS
configure port $EVENT.USER_PORT qosprofile qp7
configure qosprofile qp7 minbw $lowbw maxbw $highbw ports $EVENT.USER_PORT
endif
#
########################
#Security Configurations
########################
```

```
create log message Applying_Security_Limits
# enables Denial of Service Protection for the port
if (!$match($dosprotection,yes)) then
enable dos-protect
create log message DOS_enabled
endif
#
endif
################################
# End of USER-AUTHENTICATE block
################################
#
#
###################################
# Start of USER-UNAUTHENTICATE block
###################################
if (!$match($EVENT.NAME,USER-UNAUTHENTICATED)) then
create log message Starting_8021x_Generic_UNATUH_Module_Config
if (!$match($setQuality,yes)) then
create log message UNConfig_QOS
unconfig qosprofile ports $EVENT.USER_PORT
endif
unconfig inline-power operator-limit ports $EVENT.USER_PORT
endif
#################################
# End of USER-UNAUTHENTICATE block
#################################
create log message End_802_1x_Generic_Module_Config
```

Authentication-Triggered Avaya Profile

This script has been created for Avaya phones that support both 802.1X authentication and *LLDP*. Instead of using *DHCP* options, the phone is provisioned using LLDP parameters developed jointly by Extreme Networks and Avaya.

```
#*******************************
# Last Updated: April 11, 2007
# Tested Phones: SW4610, SW4620
# Requirements: 802.1X authentication server, VSA 203 and VSA 212 from authentiication
server. QP7 defined on the switch
#*******************************
# @MetaDataStart
# @ScriptDescription "This is a template for configuring LLDP capable Avaya phones using
the authentication trigger.  This module will provision the phone with the following
parameters: call server, file server,  dot1q, dscp, power.  Additionally the following
network side configuration is done: enable SNMP traps and QOS assignment"
# @VariableFieldLabel "Avaya phone call server IP address"
set var callserver 192.45.95.100
# @VariableFieldLabel "Avaya phone file server IP address"
set var fileserver 192.45.10.250
# @VariableFieldLabel "Send trap when LLDP event happens (true or false)"
set var sendTraps true
# @VariableFieldLabel "Set QoS Profile (true or false)"
set var setQuality true
# @MetaDataEnd
#
if (!$match($EVENT.NAME,USER-AUTHENTICATED)) then
create log message Starting_Avaya_VOIP_802.1X_AUTH_Module_Config
if (!$match($sendTraps,true)) then
enable snmp traps lldp ports $EVENT.USER_PORT
enable snmp traps lldp-med ports $EVENT.USER_PORT
else
disable snmp traps lldp ports $EVENT.USER_PORT
```

```
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
enable lldp port $EVENT.USER_PORT
configure lldp port $EVENT.USER_PORT advertise vendor-specific dot1 vlan-name
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme call-server
$callserver
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme file-server
$fileserver
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme dot1q-
framing tag
if (!$match($setQuality,true)) then
configure port $EVENT.USER_PORT qosprofile qp7
endif
endif
#
if (!$match($EVENT.NAME,USER-UNAUTHENTICATED)) then
create log message Starting_Avaya_VOIP_802.1X_UNATUH_Module_Config
if (!$match($sendTraps,true)) then
enable snmp traps lldp ports $EVENT.USER_PORT
enable snmp traps lldp-med ports $EVENT.USER_PORT
else
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
disable lldp port $EVENT.USER_PORT
if (!$match($setQuality,true)) then
unconfig qosprofile ports $EVENT.USER_PORT
endif
endif
create log message End_Avaya_VOIP_802.1X_Module_Config
```

Sample Static Profiles

The following configuration creates a profile and runs it statically:

```
* BD-10808.4 # Create upm profile p1
Enable port 1:1
.
* BD-10808.4 #run upm profile p1
* BD-10808.4 # show upm history exec 8006
UPM Profile: p1
Event:  User Request    , Time run: 2006-10-18 11:56:15
Execution Identifier: 8006      Execution Status: Pass
Execution Information:
1 # enable cli scripting
2 # set var EVENT.NAME USER-REQUEST
3 # set var EVENT.TIME 1161172575
4 # set var EVENT.PROFILE p1
5 # enable port 1:1
```

This profile creates and configures *EAPS (Extreme Automatic Protection Switching)* on the edge switch for connecting to the aggregation switch, creates specific *VLAN*s and assigns tags, configures network login, and configures the *RADIUS* client component on the switch.

```
#*******************************************
# Last Updated: May 11, 2007
# Tested Devices: SummitX EXOS 12.0
# Description: This profile configures the switch with an EAPs ring, creates specified
# vlans, configure network login, RADIUS.
```

```
#*********************************************
# @MetaDataStart
# @ScriptDescription "This is a template for configuring network parameters for edge
Summit devices.  The profile will configure the listed features:  EAPs ring, Network
login, 802.1X, vlans, and default routes."
# @VariableFieldLabel "Create EAPs ring? (yes or no)"
set var yneaps yes
# @VariableFieldLabel "Name of EAPs domain"
set var eapsdomain upm-domain
# @VariableFieldLabel "Primary port number"
set var eapsprimary 23
# @VariableFieldLabel "Secondary port number"
set var eapssecondary 24
# @VariableFieldLabel "Name of EAPs control VLAN"
set var eapsctrl upm_ctrl
# @VariableFieldLabel "Tag for EAPs control VLAN"
set var eapsctrltag 4000
# @VariableFieldLabel "Create standard VLANs? (yes or no)"
set var ynvlan yes
# @VariableFieldLabel "Name of Voice vlan"
set var vvoice voice
# @VariableFieldLabel "Voice VLAN tag"
set var vvoicetag 10
# @VariableFieldLabel "Voice VLAN virtual router"
set var vvoicevr vr-default
# @VariableFieldLabel "Name of Security Video"
set var vidsec vidcam
# @VariableFieldLabel "Security Video VLAN tag"
set var vidsectag 40
# @VariableFieldLabel "Security Video VLAN virtual router"
set var vidsecvr vr-default
# @VariableFieldLabel "Name of Data vlan"
set var vdata datatraffic
# @VariableFieldLabel "Data VLAN tag"
set var vdatatag 11
# @VariableFieldLabel "Data VLAN virtual router"
set var vdatavr vr-default
# @VariableFieldLabel "Enable Network Login? (yes or no)"
set var ynnetlogin yes
# @VariableFieldLabel "RADIUS Server IP Address"
set var radserver 192.168.11.144
# @VariableFieldLabel "RADIUS Client IP Address"
set var radclient 192.168.11.221
# @VariableFieldLabel "RADIUS Server Shared Secret"
set var radsecret goextreme
# @VariableFieldLabel "Network Login port list"
set var netloginports 1-20
# @MetaDataEnd
##################################
# Start of EAPs Configuration block
##################################
if (!$match($yneaps,yes)) then
create log message Config_EAPs
config eaps config-warnings off
create eaps $eapsdomain
config eaps $eapsdomain mode transit
config eaps $eapsdomain primary port $eapsprimary
config eaps $eapsdomain secondary port $eapssecondary
create vlan $eapsctrl
config $eapsctrl tag $eapsctrltag
config $eapsctrl qosprofile qp8
config $eapsctrl add port $eapsprimary tagged
config $eapsctrl add port $eapssecondary tagged
config eaps $eapsdomain add control vlan $eapsctrl
```

```
enable eaps
enable eaps $eapsdomain
else
create log message EAPs_Not_Configured
endif
#############
#VLAN Config
#############
if (!$match($ynvlan,yes)) then
create log message CreateStandardVLANs
create vlan $vvoice vr $vvoicevr
config vlan $vvoice tag $vvoicetag
config vlan $vvoice add port $eapsprimary tagged
config vlan $vvoice add port $eapssecondary tagged
config eaps $eapsdomain add protected $vvoice
enable lldp ports $netloginports
create qosprofile qp5
config vlan $vvoice ipa 192.168.10.221
#
create vlan $vidsec vr $vidsecvr
config vlan $vidsec tag $vidsectag
config vlan $vidsec add port $eapsprimary tagged
config vlan $vidsec add port $eapssecondary tagged
config eaps $eapsdomain add protected $vidsec
config vlan $vidsec ipa 192.168.40.221
#
create vlan $vdata vr $vdatavr
config vlan $vdata tag $vdatatag
config vlan $vdata add port $eapsprimary tagged
config vlan $vdata add port $eapssecondary tagged
config eaps $eapsdomain add protected $vdata
config vlan $vdata ipa 192.168.11.221
#  config ipr add default 192.168.11.254 vr vr-default
else
create log message NoVLANsCreated
endif
#############
#RADIUS & Netlogin
#############
if (!$match($ynnetlogin,yes)) then
create log message ConfigNetlogin
#configure $vdata ipaddress 192.168.11.221
create vlan nvlan
config netlogin vlan nvlan
config default del po $netloginports
enable netlogin dot1x
enable netlogin mac
enable netlogin ports $netloginports dot1x mac
config netlogin ports $netloginports mode mac-based-vlans
config radius netlogin primary server $radserver client-ip $radclient vr VR-Default
config radius netlogin primary shared-secret $radsecret
enable radius netlogin
config netlogin add mac-list 00:19:5B:D3:e8:DD
else
create log message NoNetlogin
endif
```

Sample Configuration with Device-Triggered Profiles

The following example demonstrates how to configure Universal Port for device detection:

```
# Create and configure the VLAN for the VoIP network.
#
switch 1 # create vlan voice
switch 2 # configure voice ipaddress 192.168.0.1/24
# Create the universal port profile for device-detect on the switch.
#
switch 3 # create upm profile detect-voip
Start typing the profile and end with a . as the first and the only character on a line.
Use - edit upm profile <name> - for block mode capability
create log message Starting_Script_DETECT-voip
set var callServer 192.168.10.204
set var fileServer 192.168.10.194
set var voiceVlan voice
set var CleanupProfile CleanPort
set var sendTraps false
#
create log message Starting_DETECT-VOIP_Port_$EVENT.USER_PORT
#*********************************************************
# adds the detected port to the device "unauthenticated" profile port list
#*********************************************************
create log message Updating_UnDetect_Port_List_Port_$EVENT.USER_PORT
configure upm event Device-UnDetect profile CleanupProfile ports $EVENT.USER_PORT
#*********************************************************
# adds the detected port to the proper VoIP vlan
#*********************************************************
configure $voiceVlan add port $EVENT.USER_PORT tag
#*********************************************************
# Configure the LLDP options that the phone needs
#*********************************************************
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme call-server
$callServer
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme file-server
$fileServer
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme dot1q-
framing tagged
configure lldp port $EVENT.USER_PORT advertise vendor-specific med capabilities
#configure lldp port $EVENT.USER_PORT advertise vendor-specific med policy application
voice vlan $voiceVlan dscp 46
#*********************************************************
# Configure the POE limits for the port based on the phone requirement
#*********************************************************
# If port is PoE capable, uncomment the following lines
#configure lldp port $EVENT.USER_PORT advertise vendor-specific med power-via-mdi
#configure inline-power operator-limit $EVENT.DEVICE_POWER ports $EVENT.USER_PORT
create log message Script_DETECT-phone_Finished_Port_$EVENT.USER_PORT
.
switch 4 #
# Create the universal port profile for device-undetect on the switch.
#
switch 5 # create upm profile clearports
Start typing the profile and end with a . as the first and the only character on a line.
Use - edit upm profile <name> - for block mode capability
create log message STARTING_UPM_Script_CLEARPORT_on_$EVENT.USER_PORT
#configure $voiceVlan delete port $EVENT.USER_PORT
unconfigure lldp port $EVENT.USER_PORT
create log message LLDP_Info_Cleared_on_$EVENT.USER_PORT
#unconfigure upm event device-undetect profile avaya-remove ports $EVENT.USER_PORT
unconfigure inline-power operator-limit ports $EVENT.USER_PORT
```

```
create log message POE_Settings_Cleared_on_$EVENT.USER_PORT
create log message FINISHED_UPM_Script_CLEARPORT_on_$EVENT.USER_PORT
.
* switch 5 #
#
# Assign the device-detect profile to the edge ports.
#
* switch 6 # config upm event device-detect profile detect-voip ports 1-10
#
# Assign the device-undetect profile to the edge ports.
#
* switch 7 # config upm event device-undetect profile clearports ports 1-10
* switch 8 #
#
# Verify that correct profiles are assigned to correct ports.
#
*  switch 9 # show upm profile
UPM Profile          Events               Flags Ports
=============================================================
clearports           Device-Undetect       e  1-10
detect-voip          Device-Detect         e  1-10
=============================================================
Number of UPM Profiles: 2
Number of UPM Events in Queue for execution: 0
Flags: d - disabled, e - enabled
Event name: log-message(Log filter name) - Truncated to 20 chars
#
# Enable LLDP message advertisements on the ports assigned to universal ports.
#
* switch 10 # enable lldp ports 1-10
```

Sample Configuration with User-Triggered Profiles

This example demonstrates how to configure a *RADIUS* server and Universal Port for user login. The first part of the example shows the RADIUS server configuration. For more information on RADIUS server configuration, see Security on page 1628.

```
# Configure the RADIUS server for the userID and password pair.
# For FreeRADIUS, edit the users file located at /etc/raddb/users as shown in the
# following lines.
#
#Sample entry of using an individual MAC addresses
00040D50CCC3    Auth-Type := EAP, User-Password == "00040D50CCC3"
Extreme-Security-Profile = "phone LOGOFF-PROFILE=clearport;",
Extreme-Netlogin-VLAN = voice
#Sample entry of using wildcard MAC addresses (OUI Method)
00040D000000    Auth-Type := EAP, User-Password == "1234"
Extreme-Security-Profile = "phone LOGOFF-PROFILE=clearport;",
Extreme-Netlogin-VLAN = voice
#Sample entry of using numeric UserID and password
10284    Auth-Type := EAP, User-Password == "1234"
Extreme-Security-Profile = "voip LOGOFF-PROFILE=voip",
Extreme-Netlogin-Vlan = voice
#Sample entry of using a text UserID and password
Sales    Auth-Type := EAP, User-Password == "Money"
Extreme-Security-Profile = "Sales-qos LOGOFF-PROFILE=Sales-qos",
Extreme-Netlogin-Vlan = v-sales
# Define the Extreme custom VSAs on RADIUS.
# For FreeRADIUS, edit the dictionary file located at //etc/raddb/dictionary to
# include the following details:
VENDOR          Extreme         1916
```

```
ATTRIBUTE       Extreme-CLI-Authorization       201     integer Extreme
ATTRIBUTE       Extreme-Shell-Command           202     string  Extreme
ATTRIBUTE       Extreme-Netlogin-Vlan           203     string  Extreme
ATTRIBUTE       Extreme-Netlogin-Url            204     string  Extreme
ATTRIBUTE       Extreme-Netlogin-Url-Desc       205     string  Extreme
ATTRIBUTE       Extreme-Netlogin-Only           206     integer Extreme
ATTRIBUTE       Extreme-User-Location           208     string  Extreme
ATTRIBUTE       Extreme-Netlogin-Vlan-Tag       209     integer Extreme
ATTRIBUTE       Extreme-Netlogin-Extended-Vlan  211     string  Extreme
ATTRIBUTE       Extreme-Security-Profile        212     string  Extreme
ATTRIBUTE       Extreme-CLI-Profile             213     string  Extreme

VALUE   Extreme-CLI-Authorization       Disabled        0
VALUE   Extreme-CLI-Authorization       Enabled         1
VALUE   Extreme-Netlogin-Only           Disabled        0
VALUE   Extreme-Netlogin-Only           Enabled         1
# End of Dictionary
# Add the switch as an authorized client of the RADIUS server.
# For FreeRADIUS, edit the file located at //etc/raddb/clients.conf to include the
# switches as details:
#
client   192.168.10.4 {
secret = purple
shortname = SummitX
# End of clients.conf
```

The rest of this example demonstrates the configuration that takes place at the ExtremeXOS switch:

```
# Create the universal port profile for user-authenticate:
* switch 1 # create upm profile phone
Start typing the profile and end with a . as the first and the only character on a line.
Use - edit upm profile <name> - for block mode capability
create log message Starting_Script_Phone
set var callServer 192.168.10.204
set var fileServer 192.168.10.194
set var voiceVlan voice
set var CleanupProfile CleanPort
set var sendTraps false
#
create log message Starting_AUTH-VOIP_Port_$EVENT.USER_PORT
#*****************************************************
# adds the detected port to the device "unauthenticated" profile port list
#*****************************************************
create log message Updating_Unauthenticated_Port_List_Port_$EVENT.USER_PORT
#*****************************************************
# Configure the LLDP options that the phone needs
#*****************************************************
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme call-server
$callServer
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme file-server
$fileServer
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme dot1q-
framing tagged
configure lldp port $EVENT.USER_PORT advertise vendor-specific med capabilities
# If port is PoE capable, uncomment the following lines
#create log message UPM_Script_A-Phone_Finished_Port_$EVENT.USER_PORT
.
switch 2 #
#
# Create the universal port profile for user-unauthenticate on the switch:
#
switch 1 # create upm profile clearport
```

```
Start typing the profile and end with a . as the first and the only character on a line.
Use - edit upm profile <name> - for block mode capability
create log message STARTING_Script_CLEARPORT_on_$EVENT.USER_PORT
unconfigure lldp port $EVENT.USER_PORT
create log message LLDP_Info_Cleared_on_$EVENT.USER_PORT
unconfigure inline-power operator-limit ports $EVENT.USER_PORT
create log message POE_Settings_Cleared_on_$EVENT.USER_PORT
create log message FINISHED_Script_CLEARPORT_on_$EVENT.USER_PORT
.
* switch 2 #
# Configure RADIUS on the edge switch.
#
* switch 4 # config radius primary server 192.168.11.144 client-ip 192.168.10.4 vr "VR-
Default"
* switch 5 # config radius primary shared-secret purple
# Configure Network Login on the edge switch.
#
For Network Login 802.1X, use the following command:
* switch 7 # create vlan nvlan
* switch 8 # config netlogin vlan nvlan
* switch 9 # enable netlogin dot1x
* switch 10 # enable netlogin ports 11-20 mode mac-based-vlans
* switch 11 # enable radius netlogin
#
# For Network Login MAC-based or OUI method, use the following command:
* switch 7 # create vlan nvlan
* switch 8 # config netlogin vlan nvlan
* switch 9 # enable netlogin mac
* switch 10 # config netlogin add mac-list 00:04:0D:00:00:00 24 1234
* switch 11 # enable radius netlogin
# Assign the user-authenticate profile to the edge port.
#
* switch 12 # configure upm event user-authenticate profile "phone" ports 11-20
* switch 13 #
# Assign the user-unauthenticate profile to the edge port.
#
* switch 14 # configure upm event user-unauthenticated profile "clearport" ports 11-20
* switch 15 #
# Check that the correct profiles are assigned to the correct ports.
#
* switch 16 # show upm profile
============================================================
UPM Profile          Events               Flags Ports
============================================================
phone                User-Authenticated      e  11-20
clearport          User-Unauthenticated      e  11-20
============================================================
Number of UPM Profiles: 5
Number of UPM Events in Queue for execution: 0
Flags: d - disabled, e - enabled
Event name: log-message(Log filter name) - Truncated to 20 chars
# Enable LLDP message advertisements on the ports.
#
* switch 17 # enable lldp ports 11-20
```

Sample Timer-Triggered Profile

The following profile and timer configuration disables *PoE* on ports 1 to 20 everyday at 6:00 p.m.:

```
* switch 1 # create upm profile eveningpoe
Start typing the profile and end with a . as the first and the only character on a line.
```

```
Use - edit upm profile <name> - for block mode capability
create log message Starting_Evening
disable inline-power ports 1-20
.
*switch 2
*switch 3 # create upm timer night
*switch 4 # config upm timer night profile eveningpoe
*switch 5 # config upm timer night at 7 7 2007 19 00 00 every 86400
```

Sample Profile with QoS Support

The example below can be used with an ExtremeSwitching series switch that supports *QoS* profiles qp1 and qp8.

When the user or phone logs in with a particular MAC address, the script configures the QoS profile configured by the user in the *RADIUS* server for the USER-AUTHENTICATED event. In this example, the user sets the QoS profile to be qp8.

You must configure network login, the RADIUS server, and Universal Port on the switch as part of the user log-in authentication process. For more information on configuring the RADIUS users file, see Security on page 1628.

The following example is an entry in the RADIUS users file for the MAC address of the phone:

```
00040D9D12A9 Auth-Type := local, User-Password == "test"
Extreme-security-profile = "p1 QOS=\"QP8\";LOGOFF-PROFILE=p2;VLAN=\"voice-        test\";"
```

Below is the Universal Port profile configuration for this example:

```
Create upm profile p1
set var z1 $uppercase($EVENT.USER_MAC)
set var z2 $uppercase(00:04:0d:9d:12:a9)
#show var z1
#show var z2
if ($match($EVENT.NAME, USER-AUTHENTICATED) == 0) then
    if ($match($z1, $z2) == 0) then
        configure port $EVENT.USER_PORT qosprofile $QOS
    endif
endif
.
```

Sample Event Profile

If not configured properly, the *STP (Spanning Tree Protocol)* can create loops in a network.

Should these loops develop, they can cause network degradation and eventually crash the network by duplicating too many Ethernet frames. By leveraging Universal Port and the Extreme Loop Recovery Protocol (ELRP) as shown in example below, it is not

only possible to detect and isolate the egress port, but it is also possible to disable the egress port to break loops.

Note

This example illustrates how to create an event profile that reconfigures the switch after an event. After this example was created, ELRP was updated with the capability to disable a port without the help of an event profile. For more information, see Using ELRP to Perform Loop Tests on page 2118.

When a loop is detected on ports where ELRP is enabled and configured, ELRP logs a message using the following format:

```
01/17/2008 08:08:04.46 <Warn:ELRP.Report.Message> [CLI:ksu:1] LOOP DETECTED : 436309
transmited, 64 received, ingress slot:port (1) egress slot:port (24)
```

To view more information on format of this command, enter the show log events command as shown in the following example:

```
* BD8810-Rack2.6 # show log events "ELRP.Report.Message" details
Comp    SubComp     Condition               Severity      Parameters
------- ----------- ----------------------- ------------- ----------
ELRP    Report      Message                 Warning        8 total
0 - string
1 - string
2 - number (32-bit
unsigned int)
3 - string
4 - number (32-bit
unsigned int)
5 - number (unsigned int)
6 - string
7 - string
"[%0%:%1%:%2%] %3% : %4% transmited, %5% received, ingress slot:port (%6%) egress
slot:port (%7%)"
```

In the example log statement, the *VLAN* ksu, the ports is all, and the interval is “1.”

If a loop is detected, we want to disable the egress port on which the ELRP control packets went out. In this example, we enable ELRP on all ports of a VLAN as follows:

```
configure elrp-client periodic vlan ports all interval 1 seconds
```

We want the profile to disable egress ports 1 and 24 (which have been configured for loop). If we enable ELRP on only one port, then the port alone would be disabled.

We observe that parameter 7 is the one we have to disable from the above log message and the details for that event.

Configuring Universal Port Example

The following procedure configures Universal Port to disable the egress port identified by parameter 7:

1. Create the profile and add the command to disable the egress port as follows:

   ```
   create upm profile disable_port_elrp
   disable port $EVENT.LOG_PARAM_7
   ```

2. Verify that the profile is created and enabled by entering the following command:

```
show upm profile
```

3. Create the EMS configuration by entering the following commands:

```
create log target upm disable_port_elrp
create log filter f1
configure log filter f1 add event ELRP.Report.Message match string “LOOP”
enable log target upm “disable_port_elrp”
configure log target upm “disable_port_elrp” filter f1
```

4. At this point, connect the ports 1 and 24 to form a loop.

   Two log messages will be logged when the loop is detected on ports 1 and 24 and ELRP is enabled both. This triggers the disable_port_elrp profile twice, and ports 1 and 24 should be disabled.

5. View the log.

```
> show log
01/17/2008 08:08:05.49 <Info:vlan.dbg.info> Port 24 link down
01/17/2008 08:08:05.22 <Noti:UPM.Msg.upmMsgExshLaunch> Launched profile
disable_port_elrp for the event log-message
01/17/2008 08:08:04.69 <Info:vlan.dbg.info> Port 1 link down
01/17/2008 08:08:04.46 <Noti:UPM.Msg.upmMsgExshLaunch> Launched profile
disable_port_elrp for the event log-message
01/17/2008 08:08:04.46 <Warn:ELRP.Report.Message> [CLI:ksu:1] LOOP DETECTED : 436309
transmited, 64 received, ingress slot:port (1) egress slot:port (24)
01/17/2008 08:08:04.46 <Warn:ELRP.Report.Message> [CLI:ksu:1] LOOP DETECTED : 436309
transmited, 63 received, ingress slot:port (24) egress slot:port (1)
01/17/2008 08:08:03.50 <Info:vlan.dbg.info> Port 24 link up at 1 Gbps speed and full-
duplex
```

6. To view the profile execution history, enter the show upm history command.

   f you want to see the more details, enter the show upm history details command to see all the profiles or display information on a specific event by entering the exec-id.

7. To view the configuration, use the show config upm and show config ems commands.

Sample Configuration for Generic VoIP LLDP

```
#*******************************
# Last Updated: March 20, 2007
# Tested Phones: Avaya 4610, 4620, 4625
# Requirements: LLDP capable devices
#*******************************
# @META_DATA_START
# @FileDescription "This is a template for configuring network parameters for VoIP phones
support LLDP but without 802.1X authentication.  The module is triggered through the
detection of an LLDP packet on the port.   The following network side configuration is
done: enable SNMP traps, QOS assignment, adjust POE reservation values based on device
requirements, add the voiceVlan to the port as tagged."
# @Description "Voice VLAN name"
set var voicevlan voice
# @Description "Send trap when LLDP event happens (true or false)"
set var sendTraps false
# @Description "Set QoS Profile (true or false)"
set var setQuality false
# @META_DATA_END
#
if (!$match($EVENT.NAME,DEVICE-DETECT)) then
create log message Starting_LLDP_Generic_Module_Config
```

```
# VoiceVLAN configuration
configure vlan $voicevlan add port $EVENT.USER_PORT tagged
#SNMP Trap
if (!$match($sendTraps,true)) then
create log message Config_SNMP_Traps
enable snmp traps lldp ports $EVENT.USER_PORT
enable snmp traps lldp-med ports $EVENT.USER_PORT
else
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
#Link Layer Discovery Protocol-Media Endpoint Discover
create log message Config_LLDP
configure lldp port $EVENT.USER_PORT advertise vendor-specific med capabilities
configure lldp port $EVENT.USER_PORT advertise vendor-specific dot1 vlan-name vlan
$voicevlan
configure lldp port $EVENT.USER_PORT advertise vendor-specific med policy application
voice vlan $voicevlan dscp 46
configure lldp port $EVENT.USER_PORT advertise vendor-specific med power-via-mdi
#Configure POE settings per device requirements
create log message Config_POE
configure inline-power operator-limit $EVENT.DEVICE_POWER ports $EVENT.USER_PORT
#QoS Profile
if (!$match($setQuality,true)) then
create log message Config_QOS
configure port $EVENT.USER_PORT qosprofile qp7
endif
endif
if (!$match($EVENT.NAME,DEVICE-UNDETECT) && $match($EVENT.DEVICE_IP,0.0.0.0)) then
create log message Starting_LLDP_Generic_UNATUH_Module_Config
if (!$match($sendTraps,true)) then
create log message UNConfig_SNMP_Traps
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
create log message UNConfig_LLDP
unconfig lldp port $EVENT.USER_PORT
if (!$match($setQuality,true)) then
create log message UNConfig_QOS
unconfig qosprofile ports $EVENT.USER_PORT
endif
unconfig inline-power operator-limit ports $EVENT.USER_PORT
endif
if (!$match($EVENT.NAME,DEVICE-UNDETECT) && !$match($EVENT.DEVICE_IP,0.0.0.0)) then
create log message DoNothing_0.0.0.0
create log message $EVENT.TIME
endif
create log message End_LLDP_Generic_Module_Config
```

Sample Configuration for Generic VoIP 802.1X

```
#*********************************************
# Last Updated: April 6, 2007
# Tested Phones: Avaya 4610, 4620, 4625
# Requirements: 802.1X capable devices, netlogin configured and enabled on deployment
ports
#*********************************************
# @META_DATA_START
# @FileDescription "This is a template for configuring network parameters for 802.1X
authenticated devices.  The module is triggered through successful authentication of the
device.  The following network side configuration is done: QOS assignment and enables DOS
protection.  When used with IP phones, phone provisioning is done through DHCP options."
```

```
# @Description "VLAN name to add to port"
set var vlan1 voice
# @Description "Set QoS Profile (yes or no)"
set var setQuality yes
# @Description "QoS Profile (0-100)"
set var lowbw 50
# @Description "QoS MAX Bandwidth (0-100)"
set var highbw 100
# @Description "Enable Denial of Service Protection (yes or no)"
set var dosprotection yes
# @META_DATA_END
#################################
# Start of USER-AUTHENTICATE block
#################################
if (!$match($EVENT.NAME,USER-AUTHENTICATED)) then
#############
#QoS Profile
#############
# Adds a QOS profile to the port
if (!$match($setQuality,yes)) then
create log message Config_QOS
configure port $EVENT.USER_PORT qosprofile qp7
configure qosprofile qp7 minbw $lowbw maxbw $highbw ports $EVENT.USER_PORT
endif
#
########################
#Security Configurations
########################
create log message Applying_Security_Limits
# enables Denial of Service Protection for the port
if (!$match($dosprotection,yes)) then
enable dos-protect
create log message DOS_enabled
endif
#
endif
################################
# End of USER-AUTHENTICATE block
```

Sample Configuration for Avaya VoIP 802.1X

```
#*******************************
# Last Updated: March 20, 2007
# Tested Phones: SW4610, SW4620
# Requirements: 802.1X authentication server, VSA 203 and VSA 212 from authentiication
server. QP7 defined on the switch#
*******************************
# @META_DATA_START
# @FileDescription "This is a template for configuring LLDP capable Avaya phones using
the authentication trigger.  This module will provision the phone with the following
parameters: call server, file server,  dot1q, dscp, power.  Additionally the following
network side configuration is done: enable SNMP traps and QOS assignment."
# @Description "Avaya phone call server IP address"
set var callserver 192.45.95.100
# @Description "Avaya phone file server IP address"
set var fileserver 192.45.10.250
# @Description "Send trap when LLDP event happens (true or false)"
set var sendTraps true
# @Description "Set QoS Profile (true or false)"
set var setQuality true
# @META_DATA_END
#
```

```
if (!$match($EVENT.NAME,USER-AUTHENTICATED)) then
create log message Starting_Avaya_VOIP_802.1X_AUTH_Module_Config
if (!$match($sendTraps,true)) then
enable snmp traps lldp ports $EVENT.USER_PORT
enable snmp traps lldp-med ports $EVENT.USER_PORT
else
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
enable lldp port $EVENT.USER_PORT
configure lldp port $EVENT.USER_PORT advertise vendor-specific dot1 vlan-name
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme call-server
$callserver
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme file-server
$fileserver
configure lldp port $EVENT.USER_PORT advertise vendor-specific avaya-extreme dot1q-
framing tag
if (!$match($setQuality,true)) then
configure port $EVENT.USER_PORT qosprofile qp7
endif
endif
#
if (!$match($EVENT.NAME,USER-UNAUTHENTICATED)) then
create log message Starting_Avaya_VOIP_802.1X_UNATUH_Module_Config
if (!$match($sendTraps,true)) then
enable snmp traps lldp ports $EVENT.USER_PORT
enable snmp traps lldp-med ports $EVENT.USER_PORT
else
disable snmp traps lldp ports $EVENT.USER_PORT
disable snmp traps lldp-med ports $EVENT.USER_PORT
endif
disable lldp port $EVENT.USER_PORT
if (!$match($setQuality,true)) then
unconfig qosprofile ports $EVENT.USER_PORT
endif
endif
create log message End_Avaya_VOIP_802.1X_Module_Config
Dynamic Security Policy
if (!$match($CLI_EVENT,USER-AUTHENTICATED) ) then
create access-list $(EVENT.DEVICE_MAC)_192_168_1_0 "ethernet-source-address
$EVENT.DEVICE_MAC ;
destination-address 192.168.1.0/24 " "permit "
create access-list $(EVENT.DEVICE_MAC)_192_168_2_0 "ethernet-source-address
$EVENT.DEVICE_MAC ;
destination-address 192.168.2.0/24 " "permit "
create access-list $(EVENT.DEVICE_MAC)_192_168_3_0 "ethernet-source-address
$EVENT.DEVICE_MAC ;
destination-address 192.168.3.0/24 " "permit "
create access-list $(EVENT.DEVICE_MAC)_smtp "ethernet-source-address $EVENT.DEVICE_MAC ;
destination-address 192.168.100.125/32 ; protocol tcp ; destination-port 25" "permit "
create access-list $(EVENT.DEVICE_MAC)_http "ethernet-source-address $EVENT.DEVICE_MAC ;
protocol tcp ; destination-port 80" "permit "
create access-list $(EVENT.DEVICE_MAC)_https "ethernet-source-address $EVENT.DEVICE_MAC ;
protocol tcp ; destination-port 443" "permit "
create access-list $(EVENT.DEVICE_MAC)_dhcp "protocol udp; destination-port 67" "permit"
create access-list $(EVENT.DEVICE_MAC)_deny "destination-address 0.0.0.0/0" "deny "
configure access-list add $(EVENT.DEVICE_MAC)_192_168_1_0 first port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_192_168_2_0 first port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_192_168_3_0 first port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_smtp first port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_http last port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_https last port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_dhcp first port $USER_PORT
configure access-list add $(EVENT.DEVICE_MAC)_deny last port $USER_PORT
```

```
endif
if (!$match($CLI_EVENT,USER-UNAUTHENTICATED) ) then
# Clean up
configure access-list delete $(EVENT.DEVICE_MAC)_192_168_1_0 ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_192_168_2_0 ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_192_168_3_0 ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_smtp ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_http ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_https ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_dhcp ports $USER_PORT
configure access-list delete $(EVENT.DEVICE_MAC)_deny ports $USER_PORT
delete access-list $(EVENT.DEVICE_MAC)_192_168_1_0
delete access-list $(EVENT.DEVICE_MAC)_192_168_2_0
delete access-list $(EVENT.DEVICE_MAC)_192_168_3_0
delete access-list $(EVENT.DEVICE_MAC)_smtp
delete access-list $(EVENT.DEVICE_MAC)_http
delete access-list $(EVENT.DEVICE_MAC)_https
delete access-list $(EVENT.DEVICE_MAC)_dhcp
delete access-list $(EVENT.DEVICE_MAC)_deny
endif
```

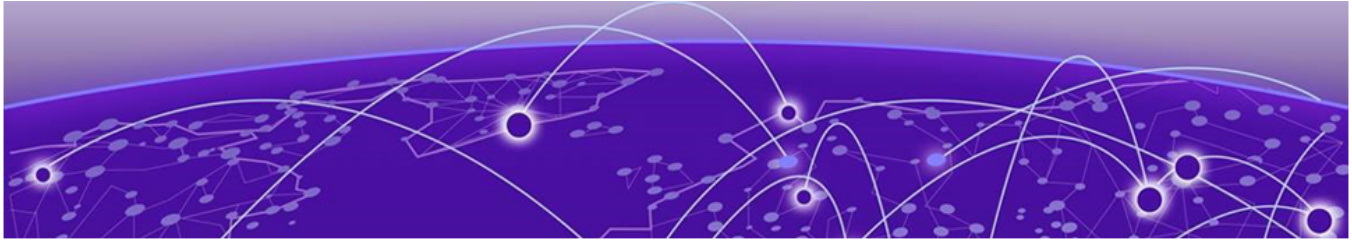
Sample Configuration for a Video Camera

This template adds an *ACL* to an edge port when a video camera connects.

The profile configures and applies an ACL onto a switch port when a user authenticates. This ACL blocks a particular IP address from accessing the video camera and assigns the user to *QoS* profile 7.

```
#*********************************************
# Last Updated: March 9, 2007
# Tested Devices: Dlink DCS 1110
# Requirements: netlogin configured and enabled on deployment ports
#*********************************************
# @MetaDataStart
# @ScriptDescription "This is a template for configuring the switch for the right
environment for this webcam. It creates a dynamic access-list to restrict access"
# @Description "VLAN name to add to port"
# set var vlan1 voiceavaya
# @VariableFieldLabel "Set QoS Profile (yes or no)"
# set var setQuality yes
# @Description "QoS Profile (0-100)"
# set var lowbw 50
# @VariableFieldLabel "QoS MAX Bandwidth (0-100)"
# set var highbw 100
# @MetaDataEnd
################################
# Start of USER-AUTHENTICATE block
################################
if (!$match($EVENT.NAME,USER-AUTHENTICATED)) then
#############
#QoS Profile
#############
# Adds a QOS profile to the port
#   if (!$match($setQuality,yes)) then
#    create log message Config_QOS
#    configure port $EVENT.USER_PORT qosprofile qp7
#    configure qosprofile qp7 minbw $lowbw maxbw $highbw ports $EVENT.USER_PORT
#  endif
#
#############
```

```
#ACL Section
############
# Adds an ACL to stop traffic to a particular address
create log message Config_ACL
create access-list webcamblock "destination-address 192.168.10.220/32" "deny"
configure access-list add webcamblock first port $EVENT.USER_PORT
#endif
#
endif
################################
# End of USER-AUTHENTICATE block
################################
#
#
###################################
# Start of USER-UNAUTHENTICATE block
###################################
if (!$match($EVENT.NAME,USER-UNAUTHENTICATED)) then
#   create log message Starting_8021x_Generic_UNATUH_Module_Config
#   if (!$match($setQuality,yes)) then
#     create log message UNConfig_QOS
#     unconfig qosprofile ports $EVENT.USER_PORT
#   endif
#   unconfigure inline-power operator-limit ports $EVENT.USER_PORT
#### remove acl
configure access-list delete webcamblock port $EVENT.USER_PORT
delete access-list webcamblock
endif
#################################
# End of USER-UNAUTHENTICATE block
#################################
create log message End_802_1x_Generic_Module_Config
```



Using CLI Scripting

Setting Up Scripts on page 2182
Displaying CLI Scripting Information on page 2195
CLI Scripting Examples on page 2197

CLI-based scripting allows you to create a list of commands that you can execute manually with a single command or automatically when a special event occurs.

CLI-based scripting supports variables and functions, so that you can write scripts that operate unmodified on multiple switches and in different environments. It allows you to significantly automate switch management.

Note
Special scripts can be used to configure the switch when it boots. For more information, see Using Autoconfigure and Autoexecute Files on page 1718.

Setting Up Scripts

Enabling and Disabling CLI Scripting

The **permanent** option enables CLI scripting for new sessions only and makes the configuration change part of the permanent switch configuration so that the scripting configuration remains the same when the switch reboots. When the command is used without the permanent option, it enables CLI scripting for the current session only. If you do not include the permanent option, CLI scripting is disabled the next time the switch boots.

- To support scripting, including the testing of script-related commands, enable scripting using the following command. CLI scripting is disabled by default.

  ```
  enable cli scripting {permanent}
  ```

  Note
  CLI scripting cannot be enabled when CLI space auto completion is enabled with the `enable cli space-completion` command.

- To disable scripting, use the following command:

  ```
  disable cli scripting {permanent}
  ```

Creating Scripts

There are two ways to create scripts.The method you choose depends on how you want to execute the script.

If you want to create a script file to configure a switch or a switch feature, and you plan to execute that script manually, you can create a script file.

If you want to create a script that is activated automatically when a device or user connects to or disconnects from a switch port, you should create the script with the Universal Port feature. The following sections provide more information on these options.

Creating a Script File

A script file is an ASCII text file that you can create with any ASCII text editor program. The text file can contain CLI commands and can use the scripting options described in the following sections:

- Using Script Variables on page 2187
- Using Special Characters in Scripts on page 2189
- Using Operators on page 2190
- Using Control Structures in Scripts on page 2191
- Using Built-In Functions on page 2192

You can move an ASCII script file to the switch using the same techniques described for managing ASCII configuration files in Software Upgrade and Boot Options on page 1697.

Creating Scripts for Use with the Universal Port Feature

The Universal Port feature allows you to create dynamic profiles that are activated by a trigger event, such as a user or device connection to a switch port.

These dynamic profiles contain script commands and cause dynamic changes to the switch configuration to enforce a policy. The universal port profiles support all the scripting options listed in Creating Scripts for Use with the Universal Port Feature on page 2183 for creating script files. For more information on entering script commands in a universal port profile, see Universal Port on page 2135.

ExtremeXOS Shell RC Script

The ExtremeXOS Shell (EXSH) will run an RC script after a successful login. This script must be named `exshrc.xsf` and placed in the default configuration directory (`/usr/local/cfg`). This script is not user-account-specific and runs when any user logs in.

This RC script follows the ExtremeXOS scripting rules and is primarily meant for shell-specific configuration, but is capable of running any ExtremeXOS commands.

The following are some examples of commands that could be placed in this file:

```
alias ps "show process"
disable clipaging
configure cli columns 256 lines 200
```

Python Scripting

Introduced in ExtremeXOS 15.6, Python scripting provides the ability to customize your switch and add functionality quickly by downloading and running scripts without the need for engineering expertise. Python scripting is extended using the synonymous `load script` and `run script` commands.

Note

To use Python scripting, FIPS mode must be off (`configure security fips-mode [on | off]` and you need to ensure that Python support is enabled (`configure security python [on | off]`).

Expect Functionality

It may be necessary to interact with ExtremeXOS using Expect scripting functionality. The Python community offers a pexpect.py module that can provide this capability. ExtremeXOS uses pexpect.py version 3.2. For documentation, see https://pypi.python.org/pypi/pexpect/

The pexpect.py provides interfaces to run interactive shell commands and to process the results through expect like functions. The challenge with exsh.clicmd() is that it is a synchronous call and not a separate process. The exshexpect.py module was developed to provide a wrapper for pexpect.py that interfaces to exsh.clicmd().

Below is an example of using pexect together with the exshexpect module.

```
import pexpect

import exshexpect
```

Create a prompt for expect that will not match any command output.

```
exosPrompt = '::--:--::'
```

Create an expect object. Pass in the prompt and the back end function to call:

```
p = exshexpect.exshspawn(exosPrompt, exsh.clicmd)
```

Use sendline to send commands to the backend function (in this case exsh.clicmd) :

```
p.sendline('show fan') <-

idx = p.expect([exosPrompt, pexpect.EOF, pexpect.TIMEOUT])

print 'idx=',idx

print 'before->',p.before

print 'after->',p.after
```

Special case for passing in a command with an additional argument described in exsh.clicmd():

```
p.send('enable debug-mode\nC211-6BB4-E6F5-B579\n')
```

In the line above, use send() and include ‘newline’ terminations for the command and the additional parameter.

This is the same as calling:

```
exsh.clicmd('enable debug-mode', True, args='C211-6BB4-E6F5-B579')
```

but using expect to process any response.

```
idx = p.expect([exosPrompt, pexpect.EOF, pexpect.TIMEOUT])
```

Creating Sockets Using Python Scripts

Creating a socket in a Python script defaults to the management interface. You can change the VR when the socket is create by using the following example:

Example: Binding the socket to VR-Default (258) for an UDP socket:

```
import IN
from socket import *
udp_sock = socket(AF_INET, SOCK_DGRAM)
udp_sock.setsockopt(SOL_SOCKET,IN.SO_BINDTODEVICE,"vrf_258")

.
.
.
# your program here
.
.
.

udp_sock.close()
```

Note
Do not use os.system('echo 2 > /proc/self/ns_id') to do this because os.system() spawns a subprocess. The subprocess will switch to vr-default, and then exit. The calling process will still be using the same VR as before.

Using VRF_258 maps the socket to VR-Default.

In cases where the socket needs to be mapped to a user VR, include the followiung API in the script to get the VR name in kernel.

Example: API to find the VRF_name for a given user VR.

```
import exsh
def vrfNameFromvrName(vrName):
  lst = exsh.clicmd("debug cli show namespace vrId | begin namespaceEntry", True)
  lines = lst.split('\n')
  # ignore first two lines
  lines.pop(0)
  lines.pop(0)
  vrfName = None
  for line in lines:
    if line == "":
    break
  name, id, peer = line.split()
  if name == vrName:
    vrfName = "vrf_" + str(256 + (int(id)))
```

```
    break
return vrfName
```

Python Scripting Examples

Example 1

This is a single line script that illustrates the `exsh.clicmd ()` called from a Python script.

```
print exsh.clicmd('show switch', True)
```

In this example, the second parameter passed to `exsh.clicmd ()` is True. This returns the CLI display output that would have normally been sent to the user. In this case, the line prints the output anyway, which is shown below.

```
SysName:          X670V-48x
SysLocation:
SysContact:       support@extremenetworks.com, +1 888 257 3000
System MAC:       00:04:96:6D:12:A7
System Type:      X670V-48x

SysHealth check:  Enabled (Normal)
Recovery Mode:    All
System Watchdog:  Enabled

Current Time:     Tue Dec 17 01:49:10 2013
Timezone:         [Auto DST Disabled] GMT Offset: 0 minutes, name is UTC.
Boot Time:        Mon Dec 16 00:28:13 2013
Boot Count:       1559
Next Reboot:      None scheduled
System UpTime:    1 day 1 hour 20 minutes 57 seconds

Current State:    OPERATIONAL
Image Selected:   secondary
Image Booted:     secondary
Primary ver:      16.1.0.3
Secondary ver:    16.1.0.3
Config Selected:  primary.cfg
Config Booted:    primary.cfg
primary.cfg       Created by ExtremeXOS version 16.1.0.3
                  255914 bytes saved on Sun Dec  8 00:04:20 2013
```

Example 2

In this example, we create a number of *VLAN (Virtual LAN)*s in the form of *`prefix tag`* and add tagged port lists to each VLAN. There are a number of methods for collecting user input and validating the data type. This example uses a simple `while True:` loop until the input is the correct type. A more robust example would also apply range checking etc... to the vlan tag(s). E.g. Create vlans with tags from 1000 to 1099 The prefix is ‘dave’, so each vlan name will have a name like ‘dave1000’, ‘dave1001’ etc... The user provided port list will be added to each vlan name as tagged ports.

```
import sys
while True:
    prefix = raw_input('enter vlan name prefix:')
    if len(prefix):
        break

while True:
```

```
    reply = raw_input('beginning vlan tag:')
    if reply.isdigit():
        minTag = int(reply)
        break

while True:
   reply = raw_input('ending vlan tag:')
   if reply.isdigit():
       maxTag = int(reply)
       break

if minTag > maxTag:
    print 'Beginning vlan id cannot be greater than ending vlan id'
    sys.exit(-1)

while True:
    portlist = raw_input('port list:')
    if len(portlist):
        break

print 'Creating vlans from ',prefix+str(minTag),'to',prefix+str(maxTag)

for vlanId in range(minTag,maxTag + 1):
    vlanName = prefix + '{0:>04d}'.format(vlanId)
    cmd = 'create vlan '+vlanName + ' tag ' + str(vlanId)
    print exsh.clicmd(cmd, True)
    cmd = 'config vlan ' + vlanName + ' add port ' + portlist + ' tag'
    print exsh.clicmd(cmd, True)
```

Note
UPM variables can be used with Python scripting. They must be passed as arguments of the script.

Using Script Variables

Variable names must be followed by white space or otherwise enclosed in parentheses.

For example: create vlan v($X)e where X is the variable. The variable created will persist through the session and will not get reset after disable/enable cli scripting.

Table 170 shows the predefined system variables that are always available for use by any script. Predefined variables are automatically set up for use before a script or profile is executed.

Note
You must enable CLI scripting before using these variables or executing a script.

Table 170: Predefined System Variables

Variable Syntax	Definition
$STATUS	Status of last command execution. Values -100 to 100 are reserved and automatically set by the software, but you can override the value with the command `return statusCode`. Common values are: 0—Successful command completion-53—Variable not found-57—WHILE depth exceeded 78—Script timeout
$CLI.USER	User who is executing this CLI.
$CLI.SESSION_TYPE	User session type.
$CLI.SCRIPT_TIME_REMAINING	When a script timeout value is configured with the `configure cli script timeout timeout` command, the system creates this variable, which returns the following values (in seconds): If no script is running, this variable returns the configured timeout value. If a script is aborted due to timeout, this variable returns the value 0. If a script finishes execution (before the timeout value is reached), this variable returns the remaining time.
$CLI.SCRIPT_TIMEOUT	When a script timeout value is configured with the `configure cli script timeout timeout` command, the system creates this variable, which returns the current timeout value. If no script is running, this variable returns the configured timeout value.

Table 171 shows the system variables that you must define before use.

Table 171: System Variables that Must Be Created

Variable Syntax	Definition

$CLI.OUT	Output of last show command. The maximum size of this variable is 1 MB. This output can be used for operations such as match and regexp. For more information on these operations, see Using Built-In Functions on page 2192.
	You must define this variable before it is used. To define this variable, enter either of the following statements:
	`set var CLI.OUT " "`
	`set var CLI.OUT 0`
	We recommend that you delete this variable after each use.

Creating Variables

You can create your own variables and change their values. When using variables, the following guidelines apply:

- Variable names are case insensitive and are limited to 32 characters.
- The variable name must be unique.
- A variable can be referenced as $X or $(X).
- If a variable already exists, it is overwritten. No error message is displayed.
- The variable expression can be a constant value, another variable, or a combination of the above with operators and functions. Operators are described in , and functions are described in .
- Only the `set var` command supports expression evaluation.
- If the variable name X contains special characters such as +-/*, then the variable needs to be enclosed in parentheses. For example: set var z ($(x) + 100).
- When you use a variable with a TCL special character in the $TCL function, the variable must be enclosed in braces. For example: set var x $TCL(string length ${CLI.USER}.
- For more information on TCL functions, see Using Built-In Functions on page 2192.

  Note
  ExtremeXOS does not consider the dot/period character as a de-limiter. This is different from standard TCL behavior, where dot/period is considered as a de-limiter.

- To create a variable and set a value for it, or to change a variable value, use the following command:
  `set var varname _expression`

  The following examples show various ways to define or change variables:

  ```
  set var x 100

  set var x ($x + 2)

  set var y ($x - 100)
  ```

- To display all variables or a specified variables, use the following command:
  `show var {varname}`

Using Special Characters in Scripts

The dollar sign ($) character and quote (") characters have special purposes in scripts.

The ($) indicates a variable, and the (") surrounds text strings. To use these characters as regular characters, precede the special character with a backslash character (\). For example:

```
set var variablename \$<varname>
set var CLI.USER “Robert \"Bob\" Smith”
```

Scripts also support quote characters within quotes.

Using Operators

Operators allow you to manipulate variables and evaluate strings.

Some operators can be used in all numeric expressions, while others are restricted to integer or string expressions. Table 172 lists the operators supported and provides comments on when they can be used. The valid operators are listed in decreasing order of precedence.

Table 172: Operators

Operator	Action	Comments
-	Unary minus	None of these operands can be applied to string operands, and the bit-wise NOT operand can be applied only to integers.
+	Unary plus	
~	Bit-wise NOT	
!	Logical NOT	
*	Multiply	None of these operands can be applied to string operands, and the remainder operand can be applied only to integers. The remainder always has the same sign as the divisor and an absolute value smaller than the divisor.
/	Divide	
%	Remainder	
+	Add	These operands are valid for any numeric operations.
-	Subtract	
<<	Left shift	These operands are valid for integer operands only. A right shift always propagates the sign bit.
>>	Right shift	
<	Boolean less	Each operator produces 1 if the condition is true, 0 otherwise. These operators can be applied to strings as well as numeric operands, in which case string comparison is used.
>	Boolean greater	
<=	Boolean less than or equal	
>=	Boolean greater than or equal	
==	Boolean equal	Each operator produces a zero or one result. These operators are valid for all operand types.
!=	Boolean not equal	
&	Bit-wise AND	This operator is valid for integer operands only.
^	Bit-wise exclusive OR	This operator is valid for integer operands only.
|	Bit-wise OR	This operator is valid for integer operands only.
&&	Logical AND	This operator produces a result of 1 if both operands are non-zero. Otherwise, it produces a result of 0. This operator is valid for numeric operands only (integers or floating-point).

Table 172: Operators (continued)

Operator	Action	Comments
||	Logical OR	This operator produces a result of 0 if both operands are zero. Otherwise, it produces a result of 1.This operator is valid for numeric operands only (integers or floating-point).
x?y:z	If-then-else (as in the C programming language)	If x evaluates to non-zero, then the result is the value of y. Otherwise the result is the value of z. The x operand must have a numeric value.

Using Control Structures in Scripts

The CLI supports the control structures described in the following sections.

Conditional Execution

```
IF (<expression>) THEN
<statements>
ELSE
<statements>
ENDIF
```

The expression must be enclosed in parentheses.

Loop While Condition is TRUE

```
WHILE (<expression>) DO
    <statements>
ENDWHILE
```

The expression must be enclosed in parentheses.

Nesting is supported up to five levels. The Ctrl-C key combination can be used to break out of any While loop(s).

The operators mentioned in Using Operators on page 2190 can be used in an expression in the set var command or in an IF or WHILE condition.

If there is incorrect nesting of an IF condition or WHILE loop, an error message appears. If a user tries to type more than five WHILE loops or five IF conditions, an error message appears. Breaking out of any number of WHILE loops always clears the WHILE condition.

Comments can be inserted by using the number sign (#).

Using Built-In Functions

Built in functions allow you to manipulate and evaluate the variables inside your script and the script output. Table 173 shows the built-in functions. Table 174 on page 2192 shows the built-in Tool Command Language (TCL) functions.

Note
ExtremeXOS uses TCL version 8.5.14.

Table 173: Built-In Functions

Syntax	Function
$MATCH(string 1, string 2)	Compares the two strings string 1 and string 2. Returns 0 if string1 matches string2. It returns -1,0, or 1, depending on whether string1 is less than, equal to, or greater than string2.
$READ(prompt)	Displays a prompt for user input and accepts input until the user presses **[Enter]** or the session times out. Replace prompt with the prompt to display to the user.
$TCL(function args)	Calls a TCL built-in function (see Table 174 on page 2192). Note that the software does not support the simultaneous operation of multiple TCL functions. For more information on TCL functions, go to http://www.tcl.tk/man/tcl8.3/TclCmd/contents.htm.
$UPPERCASE(string)	Returns the string uppercased.
$VAREXISTS(varname)	Returns zero if the specified variable does not exist. Returns non-zero if the specified variable does exist.

Table 174: Supported TCL Functions

Function	Function Type	Description
after	System related	Execute a command after a time delay.
binary	String handling	Insert and extract fields from binary strings.
clock	System related	Obtain and manipulate time.
concat	List handling	Join lists together.
expr	Math	Evaluate an expression.
join	List handling	Create a string by joining list elements together.
lindex	List handling	Retrieve an element from a list.
linsert	List handling	Insert elements into a list.
list	List handling	Create a list.
llength	List handling	Count the number of elements in a list.
lrange	List handling	Return one or more adjacent elements from a list.

Table 174: Supported TCL Functions (continued)

Function	Function Type	Description
lreplace	List handling	Replace elements in a list with new elements.
lsearch	List handling	See if a list contains a particular element.
lsort	List handling	Sort the elements of a list.
regexp	String handling	Match a regular expression against a string.
regsub	String handling	Perform substitutions based on regular expression pattern matching.
re_syntax	String handling	Syntax of TCL regular expressions.
split	List handling	Split a string into a proper TCL list.
string	String handling	Manipulate strings.

For examples of scripts that use TCL functions, see CLI Scripting Examples on page 2197.

Control Script Configuration Persistence

When a script runs, the commands within the script can make persistent changes to the switch configuration, which are saved across reboots, or it can make non-persistent changes. Non-persistent configuration changes remain part of the switch configuration only until the switch reboots.

The default setting for scripts is non-persistent. To change the script configuration persistence setting, use the following command:

```
configure cli mode [persistent | non-persistent]
```

Saving, Retrieving, and Deleting Session Variables

Session variables are the set of variables that are active for a particular session. For example, if a device is detected on a universal port and this triggers a profile (and the script commands within it), the variable values that were active when the profile started are replaced with the variable values defined in the profile. The first session is the device-undetected session, and the second session is the device-detected session. Each session has its own set of variables and values.

The software allows you to save session variables before replacing them. In the example above, this allows you to retrieve the earlier values when the port returns to the device-undetected state. Up to five variables can be saved or retrieved at a time. These variables are saved to system memory using a key, which is an ID for the saved variables. You are responsible for generating unique keys. When you want to retrieve the variables, you use the key to identify the variables to be retrieved.

- To save up to five session variables, use the following command:
  `save var key` *`key`* `[`*`var1 var2`* `…]`

- To retrieve saved session variables, use the following command:

  `load var key key [var1 var2 …]`

- To delete saved session variables, use the following command:

  `sdelete var key key`

The variables saved by the save var command are saved using the specified key and are retrievable and restored in the context that this profile was applied. They are available to rollback events like user-unauthenticate and device-undetect.

Note
For SummitStack, session variables are saved on the master ExtremeSwitching switch. To repopulate session variables after a failover event, manually execute the appropriate script.

Nesting Scripts

The ExtremeXOS software supports three levels of nested scripts. An error appears if you attempt to start a fourth-level script. The following example demonstrates nested scripts:

```
Load script x1
# Script x1 contents:
Cmd 1
Cmd 2
Load script x2
Cmd 3
Cmd 4
# Script x2 contents:
Cmd 1
Cmd 2
Load script x3
```

In the above example, Cmd x is a generic representation for any script command. Script x1 is the first level script and launches the second level script Script x2. Script x2 launches the third level script Script x3. If Script x3 contained a command to launch another script, the script would not launch the software would generate an error.

Executing Scripts

You can execute scripts by loading a script file or through the Universal Port feature.

Execute a Script File

Transfer the script file to the switch and use the `load script filename {arg1} {arg2} ... {arg9}` command.

Executing a Universal Port Script

Universal port scripts are called profiles and are executed based on several types of trigger events, including device detection and undetection and user authentication and unauthentication.

For information on how to create profiles and configure the triggers, see Universal Port on page 2135.

Configuring Error Handling

The default error handling behavior is to ignore errors. You can change options within the scripts.

To control script error handling, use the following command:

`configure cli mode scripting [`**`abort-on-error`** | **`ignore-error`**`]`

Aborting a Script

There are three ways to abort a script:

- Press **[Ctrl]** + **C** while the script is executing.
- Configure the switch to abort a script when an error occurs by using the following command:
  `configure cli mode scripting [`**`abort-on-error`** | **`ignore-error`**`]`
- Abort a script and store a status code in the $STATUS variable by using the following command:
  `return` *`statusCode`*

Displaying CLI Scripting Information

You can use the information in the following sections to display CLI scripting information.

Viewing CLI Scripting Status

The `show switch management` command displays whether or not CLI scripting is enabled, whether or not the configuration is persistent, and the CLI scripting error mode as shown in the following example:

```
# show switch management
CLI idle timeout                 : Enabled (20 minutes)
CLI max number of login attempts : 3
CLI max number of sessions       : 8
CLI paging                       : Enabled
CLI space-completion             : Disabled (this session only)
CLI configuration logging        : Disabled (without expansion)
CLI journal size                 : 100
CLI password prompting only      : Disabled
CLI display moved-keywords       : Hidden
CLI moved-keywords hidden release: 31.5
CLI RADIUS cmd authorize tokens  : 2
CLI scripting                    : Disabled (this session only)
CLI scripting error mode         : Ignore-Error (this session only)
CLI script search path           : ".:/usr/local/cfg" (this session only)
CLI persistent mode              : Persistent (this session only)
CLI prompting                    : Enabled (this session only)
CLI screen size                  : 24 Lines 80 Columns (this session only)
CLI refresh                      : Enabled
CLI history expansion            : Disabled
```

```
Image integrity checking          : Off
Current system port notation      : slot:port
Configured system port notation   : slot:port
Telnet access                     : Enabled (tcp port 23 vr all)
                                  : Access Profile : not set
SSH access                        : Disabled (Key invalid, tcp port 22 vr all)
                                  : Secure-Mode    : Off
                                  : Access Profile : not set
SSH2 idle time                    : 60 minutes
SSH2 rekey interval               : 4096 MB and no time limit
Web access                        : Enabled (tcp port 80)
                                  : Access Profile : not set
Total Read Only Communities       : 0
Total Read Write Communities      : 0

SNMP access                       : Disabled
                                  : Access Profile : not set
SNMP Notifications                : Enabled
SNMP Notification Receivers  : None
SNMP stats:     InPkts 0       OutPkts   0       Errors 0       AuthErrors 0
                Gets   0       GetNexts  0       Sets   0       Drops      0
SNMP traps:     Sent   0       AuthTraps Enabled
SNMP inform:    Sent   0       Retries   0       Failed 0
(pacman debug) Slot-1 switch_model-EXOS.2 #
```

Viewing CLI Scripting Variables

The `show var` command displays the currently defined variables and their values as shown in the following example:

```
Switch.4 # show var
----------------------------------------
Count : 3
----------------------------------------
--------------------------------------------------------------------------
variableName                      variableValue
--------------------------------  --------------------------------
CLI.SESSION_TYPE                  serial
CLI.USER                          admin
STATUS                            0
--------------------------------------------------------------------------
```

Controlling CLI Script Output

When the `load script` command is entered, the software disables CLI scripting output until the script is complete, and then CLI scripting output is enabled.

When the CLI scripting output is disabled, the only script output displayed is the `show var {varname}` command and its output. When the CLI scripting output is enabled, all script commands and responses are displayed.

Use the `enable cli scripting` **`output`** and `disable cli scripting output` commands to control what a script displays when you are troubleshooting.

CLI Scripting Examples

The following script creates 100 VLANS with IP Addresses from 10.1.1.1/16 to 10.100.1.1/16:

```
enable cli scripting
Set var count 1
while ($count < 101) do
Create vlan v$count
configure vlan v$count ipaddress 10.$(count).1.1/16
set var count ($count + 1)
endwhile
show vlan
```

The following script introduces a 60-second delay when executed:

```
set var temp $TCL(after [expr 60 *1000])
```

The following script displays the date and time:

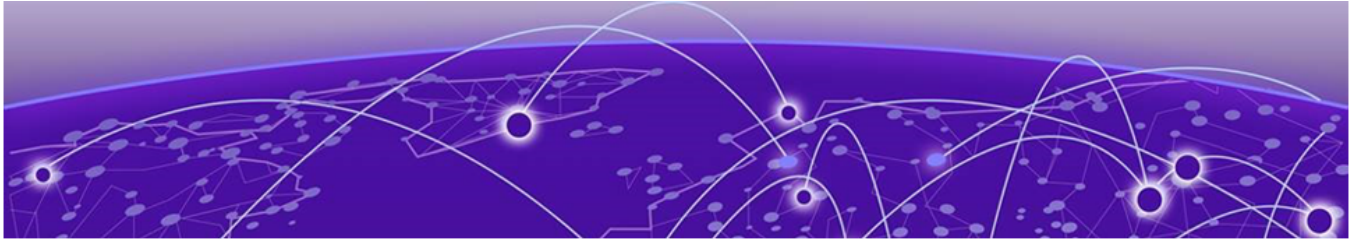
```
set var CLI.OUT " "
show switch
set var date $TCL(lrange ${CLI.OUT} 27 29)
set var year $TCL(lrange ${CLI.OUT} 31 31)
set var date $TCL(linsert $date 3 $year)
set var time $TCL(lrange ${CLI.OUT} 30 30)
show var date
show var time
```

The following script sorts the *FDB (forwarding database)* table in descending order:

```
set var CLI.OUT " "
show fdb
set var x1 $TCL(split ${CLI.OUT} "\n")
set var x2 $TCL(lsort -decreasing $x1)
set var output $TCL(join $x2 "\n")
show var output
```

The following script extracts the MAC address given the age of an FDB entry:

```
set var CLI.OUT " "
show fdb
set var input $TCL(split ${CLI.OUT} "\n")
set var y1 $TCL(lsearch -glob $input *age*)
set var y2 $TCL(lindex $input $y1)
set var y3 $TCL(split $y2 " ")
set var y4 $TCL(lindex $y3 0)
show var y4
```



Virtual Routers

Overview of Virtual Routers on page 2198
Managing Virtual Routers on page 2206
Virtual Router Configuration Example on page 2212

This section provides information about Virtual Routers. It discusses how ExtremeXOS software supports Virtual Routers and VRFs, and provides specific information about how to configure and manage those virtual routers.

Overview of Virtual Routers

The ExtremeXOS software supports virtual routers (VRs). This capability allows a single physical switch to be split into multiple VRs. This feature separates the traffic forwarded by a VR from the traffic on a different VR.

Each VR maintains a separate logical forwarding table, which allows the VRs to have overlapping IP addressing. Because each VR maintains its own separate routing information, packets arriving on one VR are never switched to another.

Note
VRs should not be connected together through a Layer 2 domain. Since there is a single MAC address per switch in the ExtremeXOS software, this same MAC address is used for all VRs. If two VRs on the same switch are connected through a Layer 2 domain, the intermediate Layer 2 switches learn the same MAC address of the switch on different ports, and may send traffic into the wrong VR.

Ports on the switch can either be used exclusively by one VR, or can be shared among two or more VRs. One reason to configure a port for the exclusive use of a single VR is to be sure that only packets from that VR egress from that port. One reason to configure a port to be shared by multiple VRs is to pass traffic from multiple VRs across a shared link.

Each *VLAN (Virtual LAN)* can belong to only one VR.

Because a single physical switch supports multiple VRs, some commands in the ExtremeXOS software require you to specify to which VR the command applies. For example, when you use the ping command, you must specify from which VR the ping packets are generated. Many commands that deal with switch management use the

management VR by default. See the Switch Engine v33.6.1 Command References for information on the defaults for individual commands.

Note
The term VR is also used with the *VRRP (Virtual Router Redundancy Protocol)*. VRRP uses the term to refer to a single that spans more than one physical router, which allows multiple switches to provide redundant routing services to users. For more information about VRRP, see VRRP Overview on page 2276.

Types of Virtual Routers

This section describes the types of Virtual Routers:

- System Virtual Routers on page 2199
- User Virtual Routers on page 2200
- VRFs on page 2201
- Local-Only Virtual Routers on page 2202
- Direct Route Leaking Between Virtual Routers on page 2203

System Virtual Routers

The system VRs are the three VRs created at boot-up time. These system VRs cannot be deleted or renamed. They are named *VR-Mgmt*, *VR-Control*, and *VR-Default*. The following describes each system VR:

- **VR-Mgmt** enables remote management stations to access the switch through Telnet, SSH, and *SNMP (Simple Network Management Protocol)* sessions; it also owns the management port. No other ports can be added to VR-Mgmt, and the management port cannot be removed from it.

  The Mgmt *VLAN* is created in VR-Mgmt during ExtremeXOS system boot-up. No other VLAN can be created in this VR, and the Mgmt VLAN cannot be deleted from it.

  No routing protocol is running on or can be added to VR-Mgmt.

  VR-Mgmt is called VR-0 in ExtremeXOS releases before 11.0.
- **VR-Control** is used for internal communications between all the subsystems in the switch. It has no external visible ports, and you cannot assign any port to it.

  VR-Control has no VLAN interface, and no VLAN can be created for it.

  No routing protocol is running on or can be added to VR-Control.

  VR-Control is called VR-1 in ExtremeXOS releases before 11.0.
- **VR-Default** is the default VR created by the ExtremeXOS system. By default, all data ports in the switch are assigned to VR-Default. Any data port can be added to and deleted from VR-Default.

Users can create and delete VLANs in VR-Default. The Default VLAN is created in VR-Default during the ExtremeXOS system boot-up. The Default VLAN cannot be deleted from VR-Default.

One instance of each routing protocol is spawned for VR-Default during the ExtremeXOS system boot-up, and these routing instances cannot be deleted.

VR-Default is called VR-2 in ExtremeXOS releases before 11.0.

User Virtual Routers

User VRs are the VRs created by users in addition to the system VRs, and each user VR supports Layer 3 routing and forwarding.

The routing tables for each VR are separate from the tables for other VRs, so user VRs can support overlapping address space.

Note
User VRs are supported only on the platforms listed for the VR feature in the Switch Engine v33.6.1 Licensing Guide document. When a SummitStack contains switches that do not support user VRs, the ports on those devices cannot be added to a user VR.

When a new user VR is created, by default, no ports are assigned, no *VLAN* interface is created, and no support for any routing protocols is added. User VRs support all switch routing protocols. When you add a protocol to a user VR, the user VR starts a process for that protocol. The ExtremeXOS software supports up to 63 user VRs, each of which supports protocols for that VR and all child Virtual Routers and Forwarding instances (VRFs).

Note
When using SNMPv2c for user created , "read community" in the *SNMP* tool should be set as "vr_name@community_name" where vr-name is user created virtual router name . Similarly for SNMPv3, "Context name" in SNMP tool should be set as "vr_name@community_name" where vr-name is user created virtual router name .

IPv4 Multicast Source Discovery Protocol on User Virtual Routers

Version 33.2.1 adds support for configuring and operating IPv4 Multicast Source Discovery Protocol (MSDP) over user-created Virtual Routers (VR). This includes the following functionality:

- Ability to configure IPv4 MSDP on IPv4 PIM-SM networks on user-defined VRs.
- MSDP peering on user VRs.
- Peer authentication using TCP MD5.
- Peer-based SA policing/filtering.
- MSDP Mesh Group support.
- MSDP Cache Server (SA Request/Response mechanism).
- Default MSDP peer and associated policy.
- Multiple default MSDP peers.

- Export of locally registered multicast sources.
- SNMP MIB access (read-only) on all scalar and tabular objects as specified by RFC-4624.

Supported Platforms

All platforms that support MSDP.

Protocol Independent Multicast Multi-switch LAG support on User-created Virtual Routers

Version 33.3.1 adds Protocol Independent Multicast (PIM) Multi-switch LAG (MLAG) support on User-created VRs. This enhancement adds support for the following features:

- IPv4 PIM-Sparse Mode (SM) on MLAG on user-created VRs.
- IPv4 PIM-Dense Mode (DM) on MLAG on user-created VRs.

The VR name must be same on both MLAG peers for PIM MLAG to work.

Supported Platforms

All platforms that support PIM MLAG.

Limitations

This following features are not supported with this enhancement:

- PIM snooping on MLAG on user-created VRs.
- IPv6 PIM features on MLAG on user-created VRs.
- PIM RP cannot be one of the MLAG peers.
- RP reachability of one MLAG peer cannot be through the other MLAG peers.

VRFs

Virtual Router and Forwarding instances (VRFs) are similar to VRs. VRFs are created as children of user VRs or *VR-Default*, and each VRF supports Layer 3 routing and forwarding. The routing tables for each VRF are separate from the tables for other VRs and VRFs, so VRFs can support overlapping address space. The primary differences between VRs and VRFs are:

- For each routing protocol added to a VRF, only one process is started in the user VR and VRF. The VRF protocol operates as one instance of the parent VR protocol, and additional child VRFs operate as additional instances of the same parent VR protocol process. VRFs allow a protocol process running in the parent VR to support many virtual router instances.
- ExtremeXOS supports up to 63 VRs and up to many more VRFs. For the maximum number of supported VRFs, see the *ExtremeXOS Release Notes*.

There are two types of VRFs:

VPN VRFs

Support BGP-based Layer 3 VPNs over *MPLS (Multiprotocol Label Switching)*. VPN VRF tables support entries for additional configuration parameters that enable Layer 3 VPN functionality over an BGP/MPLS backbone network.

Non-VPN VRFs

Support static routes, OSFPv3, and BGP.

Use VRFs instead of VRs when your network plan calls for more than 63 virtual routers or when you want to create Layer 3 VPNs. Use VRs instead of VRFs when the routing protocol you want to use is not supported on a VRF.

When a new VRF is created, by default, no ports are assigned, no *VLAN* interface is created, and no support for any routing protocols is added. When you add a protocol to a VRF, an instance of the protocol is created in the protocol process running in the parent VR, if the protocol process exists. If no protocol process is running in the parent VR, a process is started and a protocol instance corresponding to this VRF is created within that process.

The rest of this chapter uses the following terms to identify the different types of VRs and VRFs to which features and commands apply:

- VR—All VRs and VRFs
- VRF—VPN and non-VPN VRFs
- VPN VRF—VPN VRFs only
- Non-VPN VRF—Non-VPN VRFs only

Note
VRFs are supported only on the platforms listed for the VRF feature in the Switch Engine v33.6.1 Licensing Guide document. When a SummitStack contains switches that do not support VRFs, the ports on those devices cannot be added to a VRF.

Local-Only Virtual Routers

Local-only virtual routers (VR) are a form of user-created VRs (User Virtual Routers on page 2200) that allow for separate logical IP lookup tables, to be used only for IP packets to or from the switch's local IP addresses. Local-only user-created VRs allow for different gateways for routes to a host or subnet, based on the local VLANs on the switch.

Local-only VRs allow bidirectional monitoring of each individual path used to reach the switch itself, where each path may traverse, for example, a different firewall. Switches that support local-only VRs do not forward IP packets in local-only VRs in software or in hardware; they instead use separate lookup tables, including static routes, for packets to or from the switch's local IP addresses.

Supported Platforms

Local-only VRs are supported on 4120, 4220, 5120, and 5320-24T/24P/16P series switches and stacks with these switches. Other switches support VRs in hardware, rather than support local-only VRs.

Limitations

IP forwarding is not permitted on local-only VRs, or on VLANs belonging to local-only VRs.

Local-only VRs default to having IP route compression disabled. Enabling IP route compression produces an error.

Direct Route Leaking Between Virtual Routers

Version 32.4 adds support for leaking direct routes of specified VLAN(s) to a different Virtual Router (VR). Adjacent (directly connected) hosts are also reachable in the specified VR.

Note
This feature is an additional method for achieving Inter-VR routing without an external router. You can also allow a static route's gateway to be in a different VR by entering the `configure iproute add` command and specifying **`vlan`** *`egress_vlan`*, or redistribute routes from one OSPF instance to another OSPF instance in a different VR by entering the `enable ospf export {`**`vr`**`}` command.

Leaked direct routes are created with origin direct-inter-vr in the leak-to-VR. These routes have a lower route priority than direct routes and a higher route priority than any other route type.

The route priority of direct-inter-vr can be modified using the **`iproute priority`** command. These routes can be redistributed to the OSPF protocol in the leak-to-VR like any other routes in the VR. OSPF's route redistribution command is also extended as part of this feature and includes the direct-inter-vr route type.

Note
To ping a directly attached host from the switch command line successfully, the **`ping`** command must specify the correct VR name (or use the correct command line **`VR context`**) containing the VLAN with that subnet. If another VR name or VR context is used, the ping will not be successful.

Direct Route Leaking CLI Commands

Use the `configure iproute add` **`direct-inter-vr`** `{`**`ipv4`**`} [{`**`vlan`**`} `*`from_vlan_name`*` | `**`vlan`**` `*`from_vlan_list`*`] {{`**`vr`**`} `*`to_vr_name`*`}` command to add the direct routes of a VLAN or set of VLANs to be leaked to the leak-to-VR.

Use the `configure iproute delete` **`direct-inter-vr`** `{`**`ipv4`**`} [{`**`vlan`**`} `*`from_vlan_name`*` | `**`vlan`**` `*`from_vlan_list`*`] {{`**`vr`**`} `*`to_vr_name`*`}` command to delete a direct route of a VLAN or set of VLANs previously leaked to the leak-to-VR.

Use the `show iproute` **`direct-inter-vr`** `{`**`ipv4`**`} {`**`from-vr`** *`from_vr_name`*`} {{`**`vr`**`} `*`to_vr_name`*`}` command to display the configuration of Inter-VR direct routes.

Direct Route Leaking Use Case

A typical use case for this feature is when you have a printer IP directly attached to a switch on a VLAN in VR1. For example, a printer IP 10.1.1.222 on a VLAN names "yellow10" in VR1, with an IP address of 10.1.1.1/24. In this scenario, you want all users in VR1 and VR2 to access the printer using the same IP 10.1.1.222, without requireing an external router.

To accomplish this, you can configure the direct route of VLAN "yellow10" with IP 10.1.1.1/24 to be leaked to VR2. You can create a reverse routed path by configuring the direct route of VLAN "red20" with IP 20.1.1.1/24 in VR2 to be leaked to VR1.

By configuring direct route leaking, hardware and slow path forwarding tables are augmented to include all IP ARP entries on that interface, such as 10.1.1.222, in VR2 in addition to VR1.

Note
In this scenario, you do not want users in VR3 to access 10.1.1.0/24, on VR1 and VR2.

Direct Route Leaking Example Configuration

The following example commands create or configure direct route leaking.

```
create vr VR1

create vlan yellow10 tag 10 vr VR1

configure yellow10 ipaddress 10.1.1.1/24


create vr VR2

create vlan red20 tag 20 vr VR2

configure red20 ipaddress 10.1.1.1/24

configure iproute add 30.1.1.0/24 20.1.1.254 vr VR2


create vr VR3

configure iproute add direct-inter-vr vlan yellow10 vr VR2

configure iproute add direct-inter-vr red20 VR1
```

Direct Route Leaking Example Show Output

The following example displays **show** command output for this feature:

```
# show iproute direct-inter-vr

Inter-VR Direct IPv4 Routes Leaked To Virtual Router: VR1

Leak From VLAN Name              VID  Primary IP Addr.   Leak From Virtual Router
-------------------------------- ---- ------------------ -------------------------------
red20                            20   20.1.1.1       /24 VR2

Inter-VR Direct IPv4 Routes Leaked To Virtual Router: VR2

Leak From VLAN Name              VID  Primary IP Addr.   Leak From Virtual Router
-------------------------------- ---- ------------------ -------------------------------
yellow10                         10   10.1.1.1       /24 VR1
```

The following examples display the "div" origin prefix showing Direct-Inter-VR information.

```
# show iproute vr vr1

Ori  Destination        Gateway         Mtr  Flags          VLAN       Duration
 d   10.1.1.0/24        10.1.1.1        1    -------um---- yellow10   0d:0h:3m:12s
 div 20.1.1.0/24        20.1.1.1        1    -------um---- red20      0d:0h:2m:17s
 s   30.1.1.0/24        20.1.1.254      1    -G---S-um---- red20      0d:0h:1m:59s

# show iproute vr vr2
Ori  Destination        Gateway         Mtr  Flags          VLAN       Duration
 div 10.1.1.0/24        10.1.1.1        1    -------um---- yellow10   0d:0h:8m:39s
 d   20.1.1.0/24        20.1.1.1        1    -------um---- red20      0d:0h:9m:42s
 s   30.1.1.0/24        20.1.1.254      1    -G---S-um---- red20      0d:0h:0m:6s
```

VR Configuration Context

Each VR and VRF has its own configuration domain or context, in which you can enter commands to configure that VR. Some commands allow you to specify a VR to which the command applies.

For other commands, you must change context to that of the target VR or VRF before you execute the command. The current context is indicated in the command line interface (CLI) prompt by the name of the user VR or VRF. If no name appears in the prompt, the context is *VR-Default*.

For instructions on changing context, see Changing the VR Context on page 2208.

Commands that apply to the current VR context include all the *BGP (Border Gateway Protocol)*, *OSPF (Open Shortest Path First)*, *OSPFv3 (Open Shortest Path First version 3)*, PIM, IS-IS, *RIP (Routing Information Protocol)*, and *RIPng (Routing Information Protocol Next Generation)* commands, and the commands listed in the following table. Commands that apply to the current VRF context are limited to BGP commands and the commands listed in Table 175.

Table 175: Virtual Router Commands

[enable | disable] ipforwarding
configure iparp
configure iparp [**add** | **delete**]
[enable | disable] iparp
show iparp
configure iproute [**add** | **delete**]
show iproute
rtlookup
create [vlan | vman] *vlan-name*
[enable | disable] igmp
[enable | disable] igmp snooping

Table 175: Virtual Router Commands (continued)

`[enable | disable] ipmcforwarding`
`show igmp`
`show igmp snooping`
`show igmp group`
`show igmp snooping cache`

Managing Virtual Routers

This section describes the options for managing Virtual Routers.

Creating and Deleting User Virtual Routers

- To create a user VR, use the following command and do not include the type or vr attributes:

  `create virtual-router` *vr-name* {**type** [**vrf** | **vpn-vrf** {**vr** *parent_vr_name*}]}

  Note
  User VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

  A VR name cannot be the same as a VLAN name. You cannot name a user VR with the names *VR-Mgmt*, *VR-Control*, or *VR-Default* because these are the existing default system VRs. For backward compatibility, user VRs also cannot be named VR-0, VR-1 or VR-2 because these three names are the names for the system VRs in ExtremeXOS releases before 11.0.

  If you exceed the maximum number of VRs supported on your platform, a message similar to the following appears:

  ```
  Error: Maximum number of User VRs supported by the system is 63
  ```

- To create a local-only user VR (5120, 5320-24T/P, 5320-16P, 5320-24T-4X-XT, and all 4220 models only), use the following command, and do not include the type or vr attributes:

  `create virtual-router` *vr-name* **local-only** {**type** [**vrf** | **vpn-vrf** {**vr** *parent_vr_name*}]}

- To display the virtual routers, use the following command:

  `show virtual-router` {*name*}

- To delete a user VR, use the following command:

  Note
  Before you delete a VR, you must delete all VLANs and child VRFs created in that VR. All of the ports assigned to this VR are deleted and made available to assign to other VRs and VRFs. Any routing protocol that is running on the VR is shut down and deleted gracefully.

  `delete virtual-router` *vr-name*

Creating and Deleting VRFs

- To create a VRF, use the following command and include the type attribute:

  ```
  create virtual-router vr-name {type [vrf | vpn-vrf {vr
  parent_vr_name}]}
  ```

  Note
  VRFs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document. To support a Layer 3 VPN, a VPN VRF must be created under the parent VR that will run the *MPLS* protocol.

  A VRF name cannot have the same name as a VLAN or VR.
- To create a local-only VRF (5120, 5320-24T/P, 5320-16P, 5320-24T-4X-XT, and all 4220 models only), use the preceding command and include the type attribute:
- To display the VRFs, use the following command:

  ```
  show virtual-router {name}
  ```
- To delete a VRF, use the following command:

  Note
  Before you delete a VRF, you must delete all VLANs and stop all protocols that are assigned to that VRF. All of the ports assigned to a deleted VRF are deleted and made available to assign to other VRs and VRFs. Any routing protocol instance that is assigned to the VRF is deleted gracefully.

  ```
  delete virtual-router vr-name
  ```

Enabling and Disabling VRFs

VRFs are enabled when created. If you want to shut down a Layer 3 VPN, you can disable the corresponding VRF, which disables only the corresponding Layer 3 VPN and does not affect other routing services or Layer 3 VPNs.

To enable or disable a VRF, use the following commands:

```
enable virtual-router vrf-name
```

```
disable virtual-router vrf-name
```

Configuring and Removing a VR Description

A VR description is a text message that can be used to label the VR. The text message is for viewing and *SNMP* MIB reports only; it has no affect on VR operation.

To configure or remove a description, use the following commands:

```
configure vr name description string
```

```
unconfigure vr name description
```

Changing the VR Context

The VR context is introduced in VR Configuration Context on page 2205.

To switch to a context for a different VR, use the following command:

```
virtual-router {vr-name}
```

The CLI prompt displays the VR context.

Adding and Deleting Routing Protocols

When a user VR is created, no resources are allocated for routing protocols. You must add the routing protocols needed for your VR before you attempt to configure them. The maximum number of protocols supported is 64.

The commands that follow allow adding the following protocols:

- The basic seven protocols on *VR-Default* (*RIP*, *OSPF*, *BGP*, PIM, ISIS, *OSPFv3*, and RIPNG)
- 1 *MPLS* protocol instance on any VR (only on platforms that support MPLS)
- 56 additional protocols for user VRs. Any combination of the 7 protocols supported on user VRs (RIP, OSPF, BGP, PIM, ISIS, OSPFv3, and RIPNG) can be assigned to user VRs, up to a maximum number of 56.

When you add a protocol to a user VR, the software starts a process to support the protocol, but it does not enable that protocol. After you add a protocol to a user VR, you must specifically enable and configure that protocol before it starts.

When you add a protocol to a VRF, a protocol process is started in the parent VR (if it is not already started) and a protocol instance is created inside that process for this VRF.

Note
You must add, configure, and enable a protocol for a VR before you start unicast or multicast forwarding on the VR and before you configure any features (such as VLANs) to use the VR.

- To add a protocol to a VR, use the command:

  ```
  configure vr vr_name [add | delete] protocol [ospf | ospfv3 | rip | ripng | bgp | isis | pim | mpls]
  ```

  If you add more than the maximum number of protocols, the following message appears:

  ```
  Error: Maximum number of Protocols that can be started in the system is 64
  ```

- To remove a protocol from a VR, use the command:

  ```
  configure vr vr-name delete protocol protocol-name
  ```

Note
Local-only VRs and VRFs do not use routing protocols.

Configuring Ports to Use One or More Virtual Routers

By default, all the user data ports belong to *VR-Default* and the default *VLAN*, Default. All these ports are used exclusively by VR-Default. To configure a port to use one or more virtual routers, you need to perform one or more of the tasks described in the following sections:

- Deleting Ports from a Virtual Router on page 2209
- Adding Ports to a Single Virtual Router on page 2209
- Adding Ports to Multiple Virtual Routers

Deleting Ports from a Virtual Router

To configure a port for exclusive use by another VR, or for use by multiple VRs, it must first be deleted from *VR-Default*. You must delete the port from any *VLAN* it belongs to before deleting it from a VR.

To delete a port from a VR, use the command:

```
configure vr vr-name delete ports port_list
```

Caution
Do not create Layer 2 connections between ports assigned to different VRs in the same switch. Because each switch supports just one MAC address, every VR in the switch uses the same MAC address. A Layer 2 connection between two VRs can cause external devices to direct traffic to the wrong VR.

Adding Ports to a Single Virtual Router

When you add a port to a VR, that port can only be used by that VR. To add a port to a single VR, use the command:

```
configure vr vr-name add ports port_list
```

The following example demonstrates how to remove all the ports on slot 3 from the Default *VLAN* in *VR-Default* and add them for the exclusive use of user VR helix:

```
configure vlan default delete ports 3:*
configure vr vr-default delete ports 3:*
configure vr helix add ports 3:*
```

Adding Ports to Multiple Virtual Routers

To use a port in multiple VRs, do not add the port to a VR as described in Adding Ports to a Single Virtual Router on page 2209. Instead, add the port to a *VLAN* in the desired VR.

To add the port to a VLAN in the desired VR, use the following command:

```
configure [ {vlan} vlan_name | vlan vlan_list] add ports [port_list |
all] {tagged tag | untagged} {{stpd} stpd_name} {dot1d | emistp | pvst-
plus}
```

Note

See QoS for details about how multiple VRs per port can affect DiffServ and code replacement.

You should configure any protocols you want to use on a user VR before you add a VLAN to the user VR. When IP multicast forwarding will be supported on a user VR, add the PIM protocol before you enable IP multicast forwarding.

The following example demonstrates how to add port 3:5 to user VRs VR-green and VR-blue.

The tagged VLAN bldg_200 was previously configured in VR-green, and the tagged VLAN bldg_300 was previously configured in VR-blue.

```
configure vlan default delete ports 3:5
configure vr vr-default delete ports 3:5
configure vlan bldg_200 add ports 3:5 tagged
configure vlan bldg_300 add ports 3:5 tagged
```

Displaying Ports and Protocols

To display the ports, protocols, and names of protocol processes for a VR, use the following command:

```
show virtual-router {name}
```

Configuring the Routing Protocols and VLANs

After a user VR is created, the ports are added, and support for any required routing protocols is added, you can configure the VR.

- To create a *VLAN* in a VR, use the command:

  ```
  create vlan vlan_name {description vlan-description} {vr name}
  ```

  If you do not specify a VR in the create vlan command, the VLAN is created in the current VR context.

  VLAN names must conform to the guidelines specified in Object Names on page 34.

  Note

  All VLAN names and VLAN IDs on a switch must be unique, regardless of the VR in which they are created. You cannot have two VLANs with the same name, even if they are in different VRs.

- To display the VLANs in a specific VR, use the command:

  `show vlan virtual-router vr-name`, which is a specific form of this command:

  ```
  show vlan {virtual-router vr-name}
  ```

You can also configure routing protocols by using the standard ExtremeXOS software commands. The routing configurations of the different VRs are independent of each other.

Configuration Tasks for Layer 3 VPNs

Layer 3 VPN Configuration Overview

To configure VR and VRF support for a Layer 3 VPN, do the following:

1. Configure the PE to PE interfaces as follows:
   a. Select the VR that to which the *MPLS* protocol will be added.
   b. Assign PE facing VLANs and ports to the VR.
2. Configure the PE to CE interface as follows:
   a. Create a VPN VRF as described in Creating and Deleting VRFs on page 2207.
   b. Assign a VPN ID to the VPN VRF as described in Configuring a VPN ID on page 2211.
   c. Assign an RD to the VPN VRF as described in Configuring the Route Distinquisher on page 2211.
   d. Configure RTs for the VPN VRF as described in Configuring Route Targets on page 2211.
   e. Assign CE facing VLANs and ports to the VRF.
   f. Configure *BGP* as described in Configuration Overview.

Configuring a VPN ID

A VPN ID is an optional configuration parameter that you can use to associate a Layer 3 VPN ID label with a VRF.

To configure a VPN ID, use the command:

```
configure vr vrf_name vpn-id 3_byte_oui:4_vpn_index
```

Configuring the Route Distinquisher

The Route Distinquisher (RD) is added to the beginning of a VPN customer's IPv4 prefix to create a globally unique VPNv4 prefix for routing over the Layer 3 VPN. The IPv4 hosts that connect to a VRF must have unique IPv4 addresses. To eliminate duplicate IPv4 address issues between VRFs, configure a unique RD for each VRF.

To configure or unconfigure an RD, use the following commands:

```
configure vr vrf_name rd [2_byte_as_num:4_byte_number |
ip_address:2_byte_number | 4_byte_as_num:2_byte_number]

unconfigure vr vrf_name rd
```

Configuring Route Targets

Route Targets (RTs) are used by *BGP* as extended communities that define which VRFs will export and import learned routes. A VRF import RT will receive Layer 3 VPN routes from any PE VRF that is configured to export to that RT. To enable BGP Layer 3 VPN

communications, you must configure import and export RTs on every VRF that will participate in the Layer 3 VPN.

To add, delete, or configure an RT, use the command:

```
configure vr vrf_name route-target [import | export | both] [add |
delete] [route_target_extended_community]
```

Enabling and Disabling Layer 3 VPN SNMP Traps for a VR

To enable or disable Layer 3 VPN *SNMP* traps for a VR, use the following commands:

```
enable snmp trap l3vpn {vr}
```

Note
You must enable this command in the parent VR of VPN-VRF.

```
disable snmp trap l3vpn {vr}
```

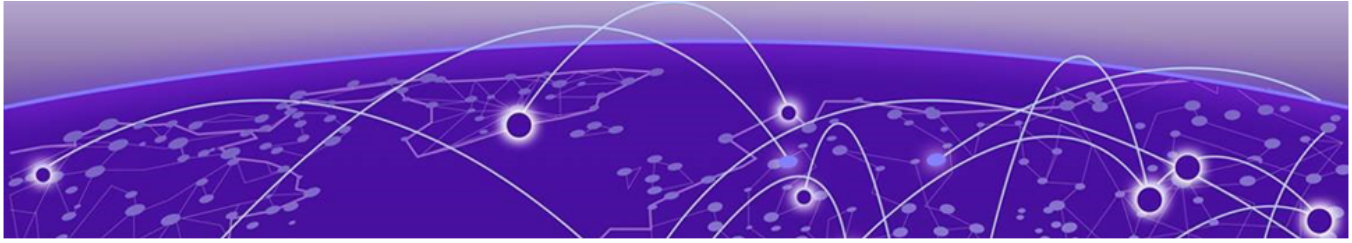
Virtual Router Configuration Example

The following example demonstrates how to:

- Create a user VR named helix.
- Remove ports from the *VLAN* Default and *VR-Default*.
- Add ports to user VR helix.
- Add the *OSPF* protocol to user VR helix.
- Set the VR context to helix, so that subsequent VR commands affect VR helix.
- Create an incoming VLAN named helix-accounting-in.
- Create an outgoing VLAN named helix-accounting-out.
- Add ports that belong to user VR helix to the helix-accounting incoming and outgoing VLANs.

The CLI prompt is shown in this example to show how the VR context appears. At the end of the example, the VR is ready to be configured for OSPF, using ExtremeXOS software commands.

```
* BD10K.1 # create virtual-router helix
* BD10K.2 # configure vlan default delete ports 3:*
* BD10K.3 # configure vr vr-default delete ports 3:*
* BD10K.4 # configure vr helix add ports 3:*
* BD10K.5 # configure vr helix add protocol ospf
* BD10K.6 # virtual-router helix
* (vr helix) BD10K.8 # configure helix-accounting-in add ports 3:1
* (vr helix) BD10K.8 # configure helix-accounting-out add ports 3:2
* (vr helix) BD10K.9 #
```



VLANs

VLANs Overview on page 2213
Configuring VLANs on the Switch on page 2223
Displaying VLAN Information on page 2229
Private VLANs on page 2230
VLAN Translation on page 2249
Port-Specific VLAN Tag on page 2257

This chapter contains information about configuring *VLAN (Virtual LAN)*s, displaying VLAN information, private VLANs, and VLAN translation. In addition, you can learn about the benefits and types of VLANs, along with valuable information about virtual routers.

VLANs Overview

Setting up *VLAN*s on the switch eases many time-consuming tasks of network administration while increasing efficiency in network operations.

Note
The software supports using IPv6 addresses, in addition to IPv4 addresses. You can configure the VLAN with an IPv4 address, IPv6 address, or both. See IPv6 Unicast Routing on page 955 for complete information on using IPv6 addresses.

The term *VLAN* is used to refer to a collection of devices that communicate as if they were on the same physical LAN.

Note
ExtremeXOS supports only 4,092 user-configurable VLANs. (VLAN 1 is the default VLAN, and 4,095 is the management VLAN, and you may not configure them.

Any set of ports (including all ports on the switch) is considered a VLAN. LAN segments are not restricted by the hardware that physically connects them. The segments are defined by flexible user groups that you create with the command line interface (CLI).

Note
The system switches traffic within each VLAN using the Ethernet MAC address. The system routes traffic between two VLANs using the IP addresses.

Benefits

Implementing *VLAN*s on your networks has the following advantages:

- **VLANs help to control traffic**—With traditional networks, broadcast traffic that is directed to all network devices, regardless of whether they require it, causes congestion. VLANs increase the efficiency of your network because each VLAN can be set up to contain only those devices that must communicate with each other.
- **VLANs provide extra security**—Devices within each VLAN can communicate only with member devices in the same VLAN. If a device in VLAN Marketing must communicate with devices in VLAN Sales, the traffic must cross a routing device.
- **VLANs ease the change and movement of devices**—With traditional networks, network administrators spend much of their time dealing with moves and changes. If users move to a different subnetwork, the addresses of each endstation must be updated manually.

Virtual Routers and VLANs

The ExtremeXOS software supports virtual routers. Each port can belong to multiple s. Ports can belong to different *VLAN*s that are in different virtual routers.

If you do not specify a virtual router when you create a VLAN, the system creates that VLAN in the default virtual router (*VR-Default*). The management VLAN is always in the management virtual router (*VR-Mgmt*).

After you create virtual routers, the ExtremeXOS software allows you to designate one of these virtual routers as the domain in which all your subsequent configuration commands, including VLAN commands, are applied. After you create virtual routers, ensure that you are creating each VLAN in the desired virtual router domain. Also, ensure that you are in the correct virtual router domain before you begin modifying each VLAN.

For information on configuring and using virtual routers, see Virtual Routers on page 2198.

Types of VLANs

This section introduces the following types of VLANs:

- Port-Based VLANs
- Tagged VLANs
- Protocol-Based VLANs
- XNV Dynamic VLAN on page 529

Note

You can have netlogin dynamic VLANs and netlogin MAC-based VLANs. See Network Login on page 1214 for complete information on netlogin.

VLANs can be created according to the following criteria:

- Physical port
- IEEE 802.1Q tag
- Ethernet, LLC SAP, or LLC/SNAP Ethernet protocol type
- A combination of these criteria

Port-Based VLANs

In a port-based *VLAN*, a VLAN name is given to a group of one or more ports on the switch.

At boot-up, all ports are members of the port-based VLAN default. Before you can add any port to another port-based VLAN, you must remove it from the default VLAN, unless the new VLAN uses a protocol other than the default protocol any. An untagged port can be a member of only one port-based VLAN.

On the Extreme Networks switch in Figure 236, ports 9 through 14 are part of VLAN Marketing; ports 25 through 29 are part of VLAN Sales; and ports 21 through 24 and 30 through 32 are in VLAN Finance.

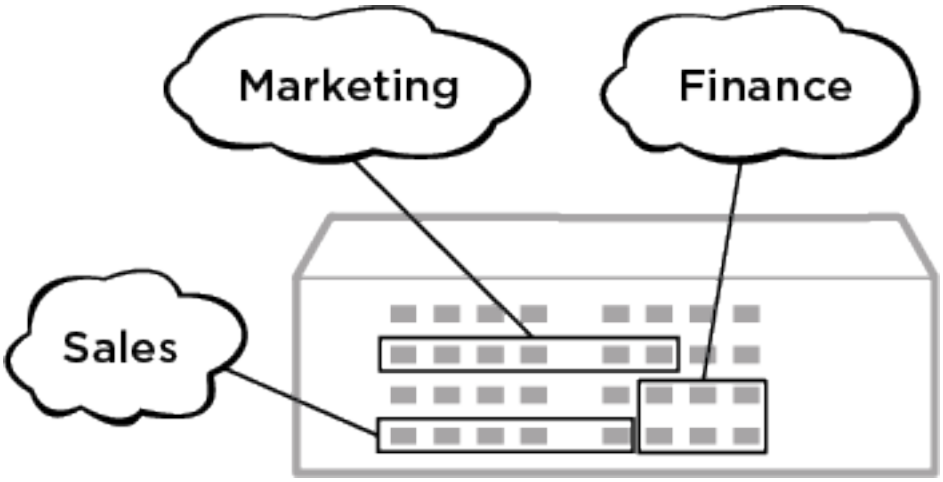


Figure 236: Example of a Port-Based VLAN on an Extreme Networks Switch

For the members of different IP VLANs to communicate, the traffic must be routed by the switch, even if the VLANs are physically part of the same switch. This means that each VLAN must be configured as a router interface with a unique IP address.

Spanning Switches with Port-Based VLANs

To create a port-based *VLAN* that spans two switches, you must do two things:

1. Assign the port on each switch to the VLAN.

2. Cable the two switches together using one port on each switch per VLAN.

   Figure 237 illustrates a single VLAN that spans a BlackDiamond switch and another Extreme Networks switch. All ports on the System 1 switch belong to VLAN Sales. Ports 1 through 29 on the system 2 switch also belong to VLAN Sales. The two switches are connected using slot 8, port 4 on System 1 (the BlackDiamond switch), and port 29 on system 2 (the other switch).

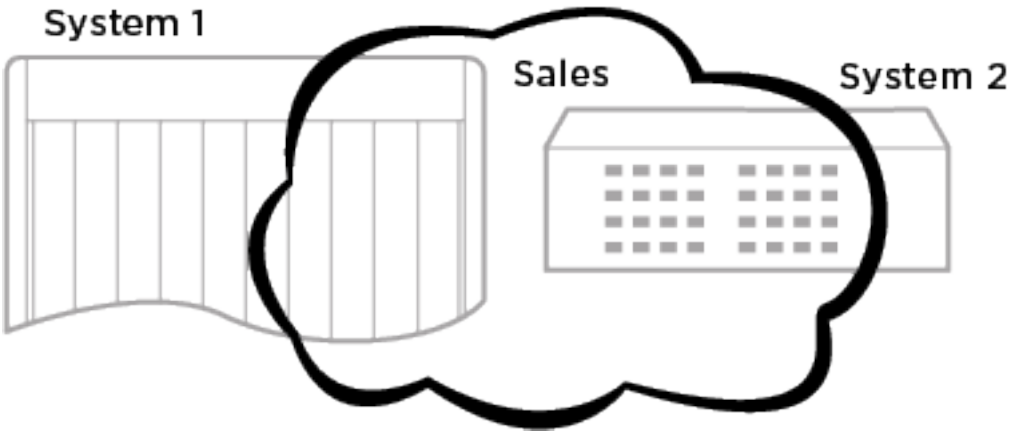


Figure 237: Single Port-based VLAN Spanning Two Switches

3. To create multiple VLANs that span two switches in a port-based VLAN, a port on System 1 must be cabled to a port on System 2 for each VLAN you want to have span across the switches.
   At least one port on each switch must be a member of the corresponding VLANs as well.

   Figure 238 illustrates two VLANs spanning two switches. On System 2, ports 25 through 29 are part of VLAN Accounting; ports 21 through 24 and ports 30 through 32 are part of VLAN Engineering. On System 1, all ports on slot 1 are part of VLAN Accounting; all ports on slot 8 are part of VLAN Engineering.

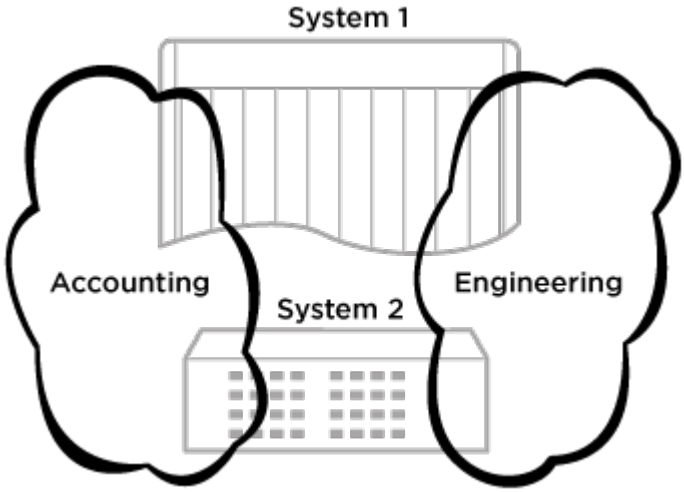


Figure 238: Two Port-Based VLANs Spanning Two Switches

   VLAN Accounting spans System 1 and System 2 by way of a connection between System 2, port 29 and System 1, slot 1, port 6. VLAN Engineering spans System 1 and System 2 by way of a connection between System 2, port 32, and System 1, slot 8, port 6.
4. Using this configuration, you can create multiple port-based VLANs that span multiple switches, in a daisy-chained fashion.

Tagged VLANs

Tagging is a process that inserts a marker (called a tag) into the Ethernet frame. The tag contains the identification number of a specific *VLAN*, called the VLANid (valid numbers are 1 to 4094).

Note
The use of 802.1Q tagged packets may lead to the appearance of packets slightly bigger than the current IEEE 802.3/Ethernet maximum of 1,518 bytes. This may affect packet error counters in other devices and may also lead to connectivity problems if non-802.1Q bridges or routers are placed in the path.

Uses of Tagged VLANs

Tagging is most commonly used to create VLANs that span switches.

The switch-to-switch connections are typically called *trunks*. Using tags, multiple VLANs can span multiple switches using one or more trunks. In a port-based VLAN, each VLAN

requires its own pair of trunk ports, as shown in Figure 239 on page 2219. Using tags, multiple VLANs can span two switches with a single trunk.

Another benefit of tagged VLANs is the ability to have a port be a member of multiple VLANs. This is particularly useful if you have a device (such as a server) that must belong to multiple VLANs. The device must have a Network Interface Card (NIC) that supports IEEE 802.1Q tagging.

A single port can be a member of only one port-based VLAN. All additional VLAN membership for the port must be accompanied by tags.

Assigning a VLAN Tag

Each VLAN may be assigned an 802.1Q VLAN tag. As ports are added to a VLAN with an 802.1Q tag defined, you decide whether each port uses tagging for that VLAN. The default mode of the switch is to have all ports assigned to the VLAN named default with an 802.1Q VLAN tag (VLANid) of 1 assigned.

Not all ports in the VLAN must be tagged. As traffic from a port is forwarded out of the switch, the switch determines (in real time) if each destination port should use tagged or untagged packet formats for that VLAN. The switch adds and strips tags, as required, by the port configuration for that VLAN.

Note
Packets arriving tagged with a VLANid that is not configured on a port are discarded.

Figure 239 illustrates the physical view of a network that uses tagged and untagged traffic.

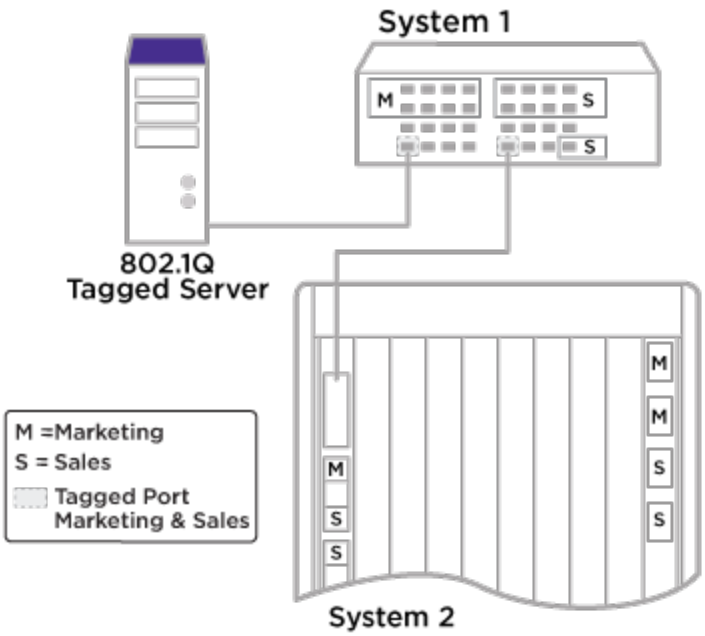


Figure 239: Physical Diagram of Tagged and Untagged Traffic

Figure 240 is a logical diagram of the same network.

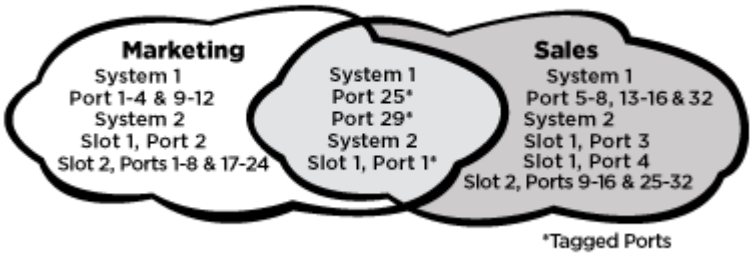


Figure 240: Logical Diagram of Tagged and Untagged Traffic

In the figures above:

- The trunk port on each switch carries traffic for both VLAN Marketing and VLAN Sales.
- The trunk port on each switch is tagged.
- The server connected to port 25 on System 1 has a NIC that supports 802.1Q tagging.
- The server connected to port 25 on System 1 is a member of both VLAN Marketing and VLAN Sales.
- All other stations use untagged traffic.

As data passes out of the switch, the switch determines if the destination port requires the frames to be tagged or untagged. All traffic coming from and going to the server is tagged. Traffic coming from and going to the trunk ports is tagged. The traffic that comes from and goes to the other stations on this network is not tagged.

Mixing Port-Based and Tagged VLANs

You can configure the switch using a combination of port-based and tagged VLANs. A given port can be a member of multiple VLANs, with the stipulation that only one of its VLANs uses untagged traffic. In other words, a port can simultaneously be a member of one port-based VLAN and multiple tag-based VLANs.

Note
For the purposes of VLAN classification, packets arriving on a port with an 802.1Q tag containing a VLANid of 0 are treated as untagged.

Protocol-Based VLANs

Protocol-based *VLAN*s enable you to define a packet filter that the switch uses as the matching criteria to determine if a particular packet belongs to a particular VLAN.

Protocol-based VLANs are most often used in situations where network segments contain hosts running multiple protocols. For example, in Figure 241, the hosts are running both the IP and NetBIOS protocols.

The IP traffic has been divided into two IP subnets, 192.207.35.0 and 192.207.36.0. The subnets are internally routed by the switch. The subnets are assigned different VLAN names, Finance and Personnel, respectively. The remainder of the traffic belongs to the VLAN named MyCompany. All ports are members of the VLAN MyCompany.

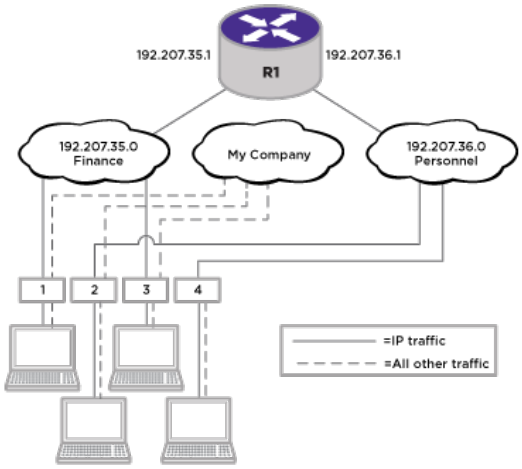


Figure 241: Protocol-Based VLANs

The following sections provide information on using protocol-based VLANs:

- Predefined Protocol Filters on page 2221
- Defining Protocol Filters on page 2221
- Configuring a VLAN to Use a Protocol Filter on page 2222
- Deleting a Protocol Filter on page 2222

Predefined Protocol Filters

The following protocol filters are predefined on the switch:

- IP (IPv4)
- IPv6 (11.2 IPv6)
- MPLS
- IPX
- NetBIOS
- DECNet
- IPX_8022
- IPX_SNAP
- AppleTalk

Defining Protocol Filters

If necessary, you can define a customized protocol filter by specifying EtherType, Logical Link Control (LLC), or Subnetwork Access Protocol (SNAP). Up to six protocols can be part of a protocol filter. To define a protocol filter:

1. Create a protocol using the following command:

   `create protocol_name`

   For example: `create protocol fred`

   The protocol name can have a maximum of 32 characters.
2. Configure the protocol using the following command:

   `configure protocol_name add [etype | llc | snap] hex {[etype | llc | snap] hex}`

   Supported protocol types include:

   etype—EtherType

   The values for etype are four-digit hexadecimal numbers taken from a list maintained by the IEEE. This list can be found at the following URL: http://standards.ieee.org/regauth/ethertype/index.html.

   Note
   Protocol-based *VLAN* for Etype from 0x0000 to 0x05ff are not classifying as per filter. When traffic arrive with these Etypes, it is classifed to native VLAN rather protocol based vlan.

   llc—LLC Service Advertising Protocol (SAP)

   The values for llc are four-digit hexadecimal numbers that are created by concatenating a two-digit LLC Destination SAP (DSAP) and a two-digit LLC Source SAP (SSAP).

   snap—EtherType inside an IEEE SNAP packet encapsulation

The values for snap are the same as the values for etype, described previously. For example:

```
configure protocol fred add llc feff
configure protocol fred add snap 9999
```

A maximum of 16 customized protocol filters can be active at a time.

Note
For more information about SNAP for Ethernet protocol types, see TR 11802-5:1997 (ISO/IEC) [ANSI/IEEE std. 802.1H, 1997 Edition].

Configuring a VLAN to Use a Protocol Filter

To configure a *VLAN* to use a protocol filter, use the following command:.

```
configure {vlan} [vlan_name |vlan_list] protocol {filter}filter_name
```

Deleting a Protocol Filter

If a protocol filter is deleted from a *VLAN*, the VLAN is assigned a protocol filter of 'any'. You can continue to configure the VLAN. However, no traffic is forwarded to the VLAN until a protocol is assigned to it.

Precedence of Tagged Packets Over Protocol Filters

If a *VLAN* is configured to accept tagged packets on a particular port, incoming packets that match the tag configuration take precedence over any protocol filters associated with the VLAN.

Default VLAN

The default switch configuration includes one default *VLAN* that has the following properties:

- The VLAN name is default.
- It contains all the ports on a new or initialized switch.
- The default VLAN is untagged on all ports. It has an internal VLANid of 1; this value is user-configurable.

VLAN Names

VLAN names must conform to the guidelines listed in Object Names on page 34.

VLAN names can be specified using the **[Tab]** key for command completion. VLAN names are locally significant. That is, VLAN names used on one switch are only meaningful to that switch. If another switch is connected to it, the VLAN names have no significance to the other switch.

Note
We recommend that you use VLAN names consistently across your entire network.

You must use mutually exclusive names for the following:

- VLANs
- VMANs
- IPv6 tunnels
- SVLANs
- CVLANs
- BVLANs

Configuring VLANs on the Switch

Refer to the following sections for instruction on configuring VLANs on a switch:

- VLAN Configuration Overview on page 2223
- Creating and Deleting VLANs on page 2224
- Managing a VLAN IP Address on page 2224
- Configuring a VLAN Tag on page 2225
- Adding and Removing Ports from a VLAN on page 2225
- Adding and Removing VLAN Descriptions on page 2226
- Renaming a VLAN on page 2226
- Enabling and Disabling VLANs on page 2226
- VLAN Configuration Examples on page 2227

VLAN Configuration Overview

The following procedure provides an overview of *VLAN* creation and configuration:

1. Create and name the VLAN.
   `create [ {vlan} vlan_name | vlan vlan_list] {tag tag } {description vlan-description } {vr name }`
2. If needed, assign an IP address and mask (if applicable) to the VLAN as described in Managing a VLAN IP Address on page 2224.
3. If any ports in this VLAN will use a tag, assign a VLAN tag.
   `configure {vlan} vlan_name tag tag {remote-mirroring}`
4. Assign one or more ports to the VLAN.

   As you add each port to the VLAN, decide if the port will use an 802.1Q tag.
5. For the management VLAN on the switch, configure the default IP route for *VR-Mgmt*.

   Note
   See IPv4 Unicast Routing for information on configuring default IP routes or adding secondary IP addresses to VLANs.

VLAN ID and VLAN ID List Specification

As of ExtremeXOS 16.1, you can now refer to a *VLAN* by VID as an alternative. Specifying lists of VIDs is a is a useful shortcut to all users that can reduce the number of commands required to configure the switch.

For example consider a case where two ports should be added to four tagged VLANs. Normally this would require four commands:

```
configure vlan red add ports 2,10 tagged

configure vlan blue add ports 2,10 tagged

configure vlan green add ports 2,10 tagged

configure vlan orange add ports 2,10 tagged
```

Assuming the tags for the VLANs are 100, 200, 300, 400 respectively this configuration can be accomplished with a single command:

```
configure vlan 100,200,300,400 add ports 2, 10 tagged
```

Note
Commands enhanced with VID list support operate in a “best effort” fashion. If one of the VIDs in a VID list do not exist the command is still executed for all of the VIDs in the list that do exist. No error or warning is displayed for the invalid VIDs unless all of the specified VIDs are in valid.

Creating and Deleting VLANs

- To create a *VLAN*, use the following command:

  `create [ {vlan} vlan_name | vlan vlan_list] {tag tag } {description vlan-description } {vr name }`

- To delete a VLAN, use the following command:

  `delete vlan vlan_name | vlan_list`

Managing a VLAN IP Address

Note
If you plan to use this *VLAN* as a control VLAN for an *EAPS (Extreme Automatic Protection Switching)* domain, do not assign an IP address to the VLAN.

- Configure an IP address and mask for a VLAN.

  Note
  IPv4 addresses with 31-bit prefixes can be configured on network VLANs and the management VLAN. Applications and protocols can use these IPv4 addresses.

```
configure [ {vlan} vlan_name | vlan vlan_id] ipaddress [ipaddress
{netmask} | {ipNetmask} | ipv6-link-local | {eui64} ipv6_address_mask]
```

Note

Each IP address and mask assigned to a VLAN must represent a unique IP subnet. You cannot configure the same IP subnet on different VLANs on the same .

The software supports using IPv6 addresses, in addition to IPv4 addresses. You can configure the VLAN with an IPv4 address, IPv6 address, or both. See IPv6 Unicast Routing on page 955 for complete information on using IPv6 addresses.

- Remove an IP address and mask for a VLAN.

  ```
  unconfigure [ {vlan} vlan_name | vlan vlan_id]ipaddress
  {ipv6_address_mask}
  ```

Configuring a VLAN Tag

To configure a *VLAN*, use the following command:

```
configure {vlan} vlan_name tag tag {remote-mirroring}
```

Adding and Removing Ports from a VLAN

- To add ports to a *VLAN*, use the following command:

  The system returns the following message if the ports you are adding are already *EAPS* primary or EAPS secondary ports:

  ```
  WARNING: Make sure Vlan1 is protected by EAPS, Adding EAPS ring ports to a VLAN could
  cause a loop in the network.
  Do you really want to add these ports? (y/n)
  ```

  The system returns the following message if the ports that you are adding are untagged ports already associated with an untagged VLAN, and you are adding them to a different untagged VLAN or tagged VLAN

  ```
  Port 1 untagged has been auto-moved from VLAN "Default" to "v2".
  ```

  You can enable direct movement of untagged ports by enabling auto-move globally with the `configure vlan untagged-ports auto-move [on | off |inform]` command.
- To remove ports from a VLAN, use the following command:

Adding and Removing VLAN Descriptions

A *VLAN* description is a string of up to 64 characters that you can configure to describe the VLAN. It is displayed by several show vlan commands and can be read by using *SNMP (Simple Network Management Protocol)* to access the VLAN's ifAlias MIB object.

- To add a description to a VLAN, use the following command:

  `configure {vlan} vlan_name description [vlan-description | none]`

- To remove a description from a VLAN, use the following command

  `unconfigure {vlan} vlan_name description`

Renaming a VLAN

To rename an existing *VLAN*, use the following command:

`configure {vlan} vlan_name | vlan_id name name`

The following rules apply to renaming VLANs:

- You cannot change the name of the default VLAN.
- You cannot create a new VLAN named default.

Enabling and Disabling VLANs

You can enable or disable individual *VLAN*s. The default setting is that all VLANs are enabled.

Consider the following guidelines before you disable a VLAN:

- Disabling a VLAN stops all traffic on all ports associated with the specified VLAN.
- You cannot disable any VLAN that is running any Layer 2 protocol traffic.

  When you attempt to disable a VLAN running Layer 2 protocol traffic (for example, the VLAN Accounting), the system returns a message similar to the following:

  ```
  VLAN accounting cannot be disabled because it is actively used by an L2 Protocol
  ```

- You can disable the default VLAN; ensure that this is necessary before disabling the default VLAN.
- You cannot disable the management VLAN.
- You cannot bind Layer 2 protocols to a disabled VLAN.
- You can add ports to and delete ports from a disabled VLAN.

1. Disable a VLAN by running:

   `disable vlan vlan_name | vlan_list`

2. After you have disabled a VLAN, re-enable that VLAN.

   `enable vlan vlan_name | vlan_list`

VLAN Configuration Examples

Note
To add an untagged port to a *VLAN* you create, you must first delete that port from the default VLAN. If you attempt to add an untagged port to a VLAN before deleting it from the default VLAN, you see the following error message:
`Error: Protocol conflict when adding untagged port 1:2. Either add this port as tagged or assign another protocol to this VLAN.`

The following example creates a port-based VLAN named accounting:

```
create vlan accounting
configure accounting ipaddress 132.15.121.1
configure default delete port 2:1-2:3,2:6,4:1,4:2
configure accounting add port 2:1-2:3,2:6,4:1,4:2
```

Note
Because VLAN names are unique, you do not need to enter the keyword vlan after you have created the unique VLAN name. You can use the VLAN name alone (unless you are also using this name for another category such as *STPD (Spanning Tree Domain)* or *EAPS*, in which case we recommend including the keyword vlan).

The following stand-alone switch example creates a port-based VLAN named development with an IPv6 address:

```
create vlan development
configure development ipaddress 2001:0DB8::8:800:200C:417A/64
configure default delete port 1-3
configure development add port 1-3
```

The following example creates a protocol-based VLAN named ipsales.

Slot 5, ports 6 through 8, and slot 6, ports 1, 3, and 4-6 are assigned to the VLAN. In this example, you can add untagged ports to a new VLAN without first deleting them from the default VLAN, because the new VLAN uses a protocol other than the default protocol.

```
create vlan ipsales
configure ipsales protocol ip
configure ipsales add port 5:6-5:8,6:1,6:3-6:6
```

The following example defines a protocol filter, myprotocol and applies it to the VLAN named myvlan. This is an example only, and has no real-world application.

```
create protocol myprotocol
configure protocol myprotocol add etype 0xf0f0
configure protocol myprotocol add etype 0xffff
create vlan myvlan
configure myvlan protocol myprotocol
```

To disable the protocol-based VLAN (or any VLAN) in the above example, use the following command:

```
disable vlan myprotocol
```

To re-enable the VLAN, use the following command:

```
enable vlan myprotocol
```

IP and MAC Anycast

The IP and MAC anycast feature enables you to configure the anycast gateway MAC to be used by VLANs that enable IP anycast. Using this feature, you can set the same IP address and MAC address on all edge technology devices. This allows seamless IP mobility in the network for edge devices.

Note
IP and MAC anycast should be used only with VXLAN tenant VLANs. IP and MAC anycast should not be used with any protocols other than MLAG, since it might cause MAC flaps.

Note
Switching through a VXLAN tunnel to a remote L3 Anycast gateway is not supported.

To configure IP/MAC anycast:

1. Enable IP anycast on the desired VLANs by using the following command:

   `enable` **`ip anycast`** {**`vlan`**} *`vlan_name`*

2. Specify the anycast gateway MAC address that is used by VLANs that enable IP anycast by using the following command:

   `configure` **`ip anycast mac`** [**`none`** | *`mac`*]

3. To specify that configured secondary IP addresses respond to ARP requests with the anycast MAC address, use the following command and specify the anycast option:

   `configure [ {`**`vlan`**`} vlan_name | `**`vlan`** *`vlan_id`*]**`add secondary-ipaddress anycast`** [*`ip_address`* {*`netmask`*} | *`ipNetmask`*]

4. To configure a source VLAN to use as the source IP address in the BOOTPrelay packet, use the following command:

   `configure` **`bootprelay dhcp-agent source-vlan`** {*`vlan_name`*} {**`vr`** *`vrid`*}

5. To enable or disable a reply to multicast or anycast echo requests, use the following commands:

   ```
   enable icmp ipv6 [ignore-multicasts | ignore-anycasts]

   disable icmp ipv6 [ignore-multicasts | ignore-anycasts]
   ```

To view IP and MAC anycast information:

- For IP anycast status (enabled or disabled), use the following command:

```
show vlan {virtual-router vr-name}
```

- For IP anycast MAC information, use the following command:

  ```
  show ipconfig {ipv4} {vlan vlan_name | tunnel tunnel_name}
  ```

- For IPv6 anycast MAC information, use the following command:

  ```
  show ipconfig ipv6 {vlan vlan_name | tunnel tunnel_name}
  ```

- To view configured secondary IP addresses on a VLAN, use the following command:

  ```
  show vlan {virtual-router vr-name}
  ```

- To view the selected source VLAN for BOOTPrelay, use the command `show bootprelay configuration {ipv4 | ipv6} {{vlan vlan_name } | {vr vr_name}}` .
- To view configuration information for ICMP multicast and anycast echo requests, use the following command:

  ```
  show ipconfig ipv6
  ```

Displaying VLAN Information

To display general *VLAN* settings and information, use the following commands:

- `show vlan {virtual-router vr-name}`
- `show vlan vlan_name {ipv4 | ipv6}`
- `show vlan [tag tag | detail] {ipv4 | ipv6}`
- `show vlan description`
- `show vlan {vlan_name | vlan_list} statistics {no-refresh | refresh }`

> **Note**
> To display IPv6 information, you must use either the show vlan detail command or show vlan command with the name of the specified VLAN.

To display the VLAN information for other ExtremeXOS software features, use the following commands:

- `show {vlan} vlan_name dhcp-address-allocation`
- `show {vlan} vlan_name dhcp-config`
- `show {vlan} vlan_name eaps`
- `show {vlan} vlan_name security`
- `show {vlan} {vlan_name | vlan_list} stpd`

You can display additional useful information on VLANs configured with IPv6 addresses by issuing the command:

```
show ipconfig ipv6 vlan vlan_name
```

To isplay protocol information, issue the command:

```
show protocol {name}
```

Private VLANs

The following sections provide detailed information on private VLANs:

- PVLAN Overview on page 2230
- Configuring PVLANs on page 2239
- Displaying PVLAN Information on page 2242
- PVLAN Configuration Example 1 on page 2243
- PVLAN Configuration Example 2 on page 2245

PVLAN Overview

PVLANs offer the following features:

- VLAN translation
- VLAN isolation

Note
PVLAN features are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

VLAN Translation in a PVLAN

VLAN translation provides the ability to translate the 802.1Q tags for several VLANs into a single VLAN tag. VLAN translation is an optional component in a PVLAN.

VLAN translation allows you to aggregate Layer 2 VLAN traffic from multiple clients into a single uplink VLAN, improving VLAN scaling. Figure 242 shows an application of VLAN translation.

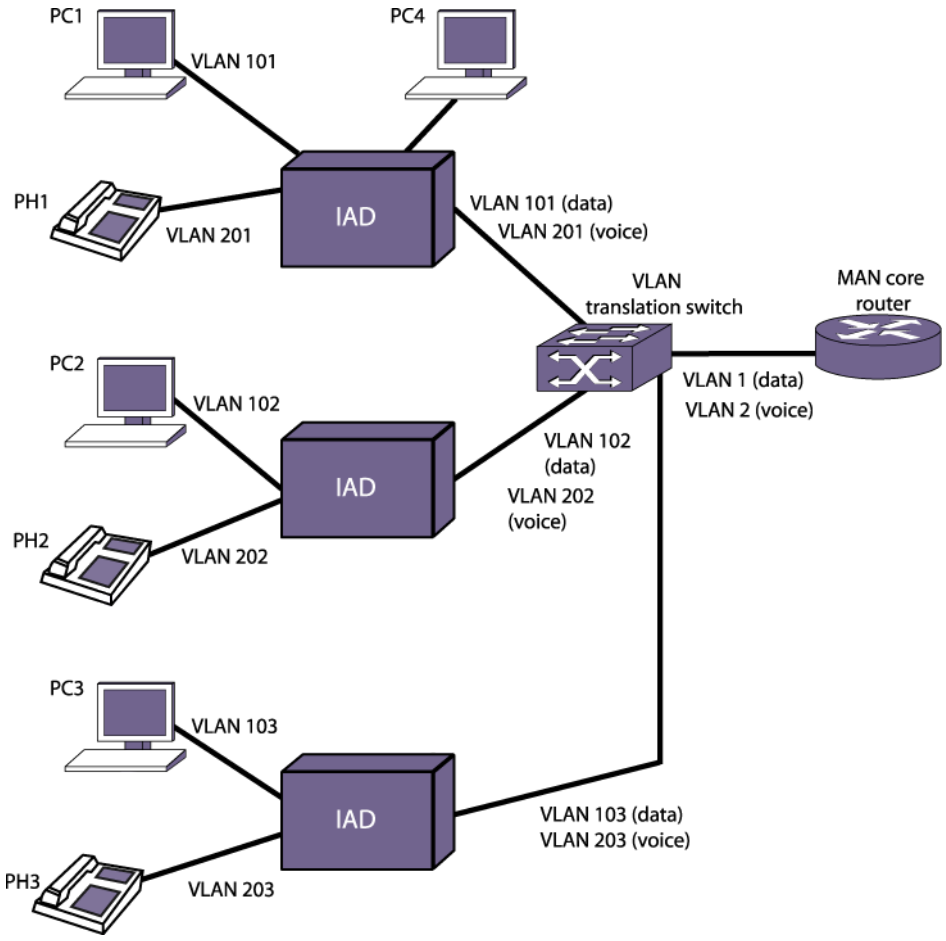


Figure 242: VLAN Translation Application

In the figure, VLANs 101, 102, and 103 are subscriber VLANS that carry data traffic while VLANs 201, 202, and 203 are subscriber VLANs that carry voice traffic. The voice and data traffic are combined on integrated access devices (IADs) that connect to the VLAN translation switch. Each of the three clusters of phones and PCs uses two VLANs to separate the voice and data traffic. As the traffic is combined, the six VLANs are translated into two network VLANs, VLAN1 and VLAN2. This simplifies administration, and scales much better for large installations.

Conceptually, this is very similar to Layer 3 VLAN aggregation (superVLANS and subVLANs).

The primary differences between these two features are:

- VLAN translation is strictly a Layer 2 feature.
- VLAN translation does not allow communication between the subscriber VLANs.

VLAN Isolation

VLAN isolation provides Layer 2 isolation between the ports in a VLAN. Figure 243 shows an application of VLAN isolation.

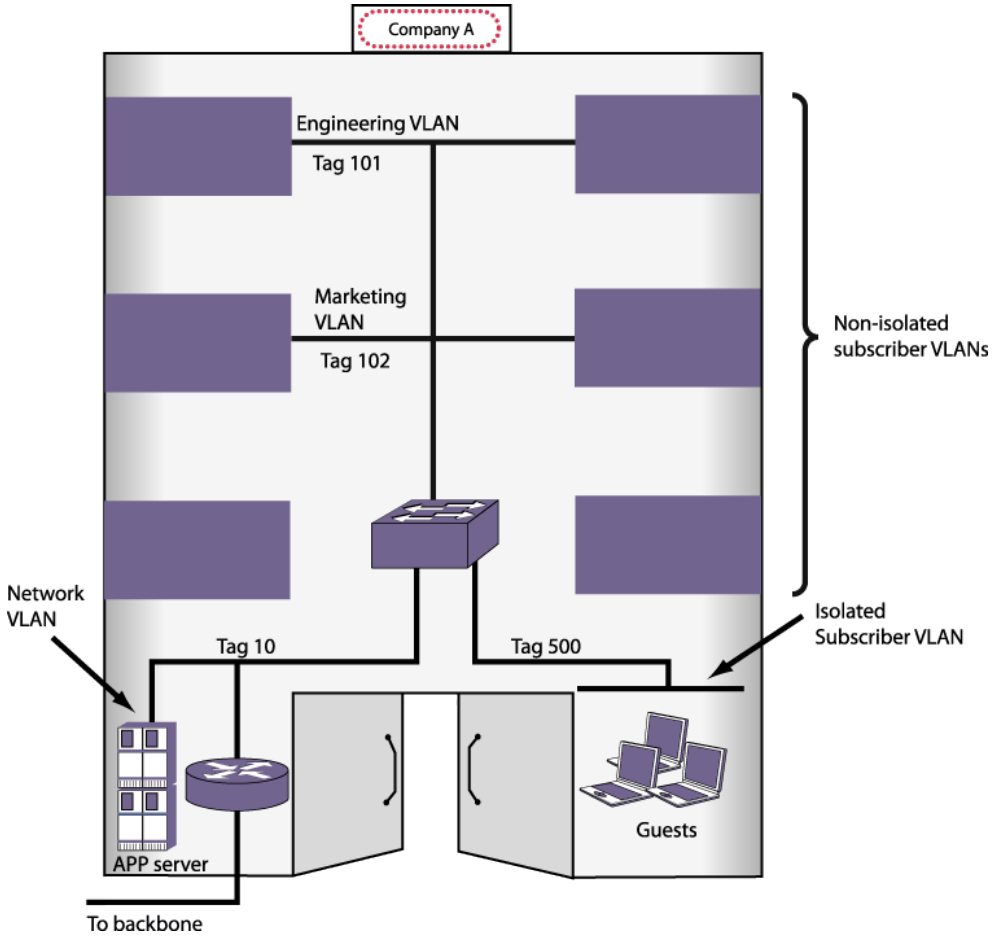


Figure 243: VLAN Isolation Application

In this figure, ports in the Guest VLAN have access to services on the network VLAN, but Guest VLAN ports cannot access other Guest VLAN ports over Layer 2 (or the Marketing or Engineering VLANs). This provides port-to-port security at Layer 2.

PVLAN Components

Figure 244 shows the logical components that support PVLAN configuration in a switch.

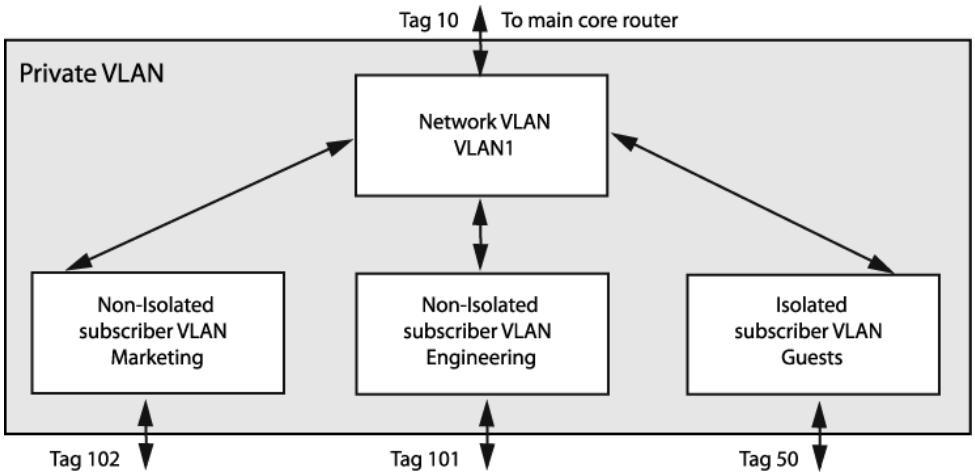


Figure 244: Private VLAN Switch Components

There is one network *VLAN* in each PVLAN. Ports within a network VLAN, called network ports, can communicate with all VLAN ports in the PVLAN. Network devices that connect to the network VLAN ports are considered to be on the network side of the switch.

The network VLAN aggregates the uplink traffic from the other VLANS, called subscriber VLANs, for egress communications on a network VLAN port. A network port can serve only one PVLAN, but it can serve one or more subscriber VLANs. Ingress communications on the network VLAN port are distributed to the appropriate subscriber VLANs for distribution to the appropriate ports. Devices that connect to subscriber VLAN ports are considered to be on the subscriber side of the switch.

Note
PVLAN network-tagged packets are allowed to ingress on subscriber VLAN ports.

Tag translation within the PVLAN is managed at the egress ports. To enable tag translation for uplink traffic from the subscriber VLANs, you must enable tag translation on the appropriate network VLAN port. Tag translation is automatically enabled on subscriber VLAN egress ports when the subscriber VLAN is created and the port is added to the VLAN as tagged. Egress traffic from a subscriber VLAN is always tagged with the subscriber VLAN tag when the port is configured as tagged.

A non-isolated subscriber VLAN is basically a standard VLAN that can participate in tag translation through the network VLAN when VLAN translation is enabled on the network VLAN port.

You can choose to not translate tags on a network VLAN port, but this is generally used only for extending a PVLAN to another switch. A non-isolated subscriber VLAN that does not use tag translation is functionally equivalent to a regular VLAN, so it is better to create non-isolated VLANs only when you plan to use tag translation.

Ports in a non-isolated VLAN can communicate with other ports in the same VLAN, ports in the network VLAN, and destinations on the network side of the switch. As with

standard VLANs, non-isolated ports cannot communicate through Layer 2 with ports in other subscriber VLANs.

In Figure 244 on page 2233, the Engineering and Marketing VLANs are configured as non-isolated subscriber VLANs, which means that they act just like traditional VLANs, and they can participate in tag translation when VLAN translation is enabled on a network VLAN port that leads to network side location.

VLAN isolation within the PVLAN is established by configuring a VLAN to be an isolated subscriber VLAN and adding ports to the isolated VLAN. Unlike normal VLANs, ports in an isolated VLAN cannot communicate with other ports in the same VLAN over Layer 2 or Layer 3. The ports in an isolated VLAN can, however, communicate with Layer 2 devices on the network side of the PVLAN through the network VLAN. When the network VLAN egress port is configured for tag translation, isolated VLAN ports also participate in uplink tag translation. When isolated subscriber VLAN ports are configured as tagged, egress packets are tagged with the isolated VLAN tag. As with standard VLANs and non-isolated VLANs, isolated ports cannot communicate through Layer 2 with ports in other subscriber VLANs.

PVLAN Support over Multiple Switches

A PVLAN can span multiple switches. Figure 244 on page 2233 shows a PVLAN that is configured to operate on two switches.

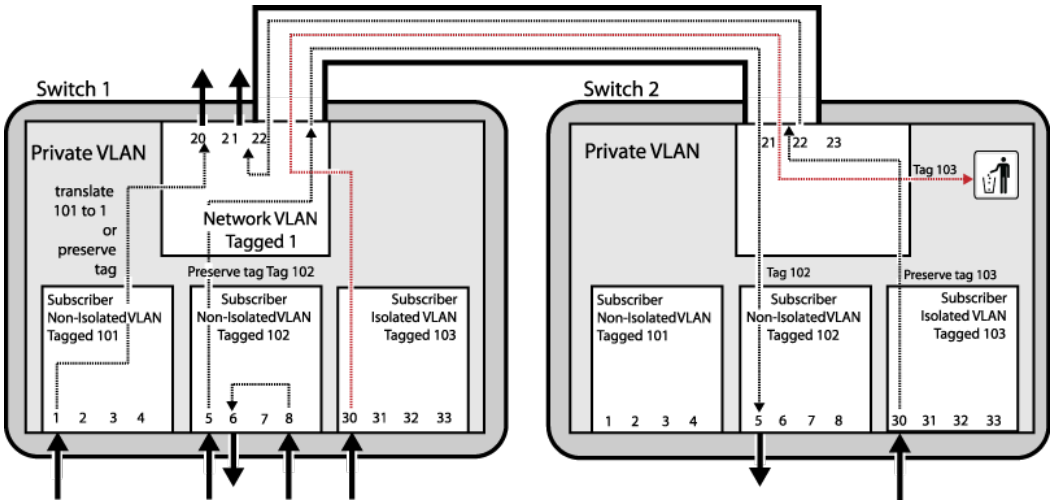


Figure 245: Private VLAN Support on Multiple Switches

A PVLAN can span many switches. For simplicity, this example shows only two switches, but you can extend the PVLAN to additional switches by adding connections between the network VLANs in each switch. The ports that connect two PVLAN switches must be configured as regular tagged ports. The network and subscriber VLANs on each switch must be configured with the same tags.

Note
Although using the same VLAN names on all PVLAN switches might make switch management easier, there is no software requirement to match the VLAN names. Only the tags must match.

When a PVLAN is configured on multiple switches, the PVLAN switches function as one PVLAN switch. Subscriber VLAN ports can access the network VLAN ports on any of the PVLAN switches, and non-isolated VLAN ports can communicate with ports in the same VLAN that are located on a different physical switch. An isolated VLAN can span multiple switches and maintain isolation between the VLAN ports.

The network and subscriber VLANs can be extended to other switches that are not configured for the PVLAN (as described in Extending Network and Subscriber VLANs to Other Switches on page 2235). The advantage to extending the PVLAN is that tag translation and VLAN isolation is supported on the additional switch or switches.

Extending Network and Subscriber VLANs to Other Switches

A network or subscriber *VLAN* can be extended to additional switches without a PVLAN configuration on the additional switches.

You might want to do this to connect to existing servers, switches, or other network devices. You probably do not want to use this approach to support clients, as tag translation and VLAN isolation are not supported unless the PVLAN is configured on all PVLAN switches as described in PVLAN Support over Multiple Switches on page 2234.

Figure 246 illustrates PVLAN connections to switches outside the PVLAN.

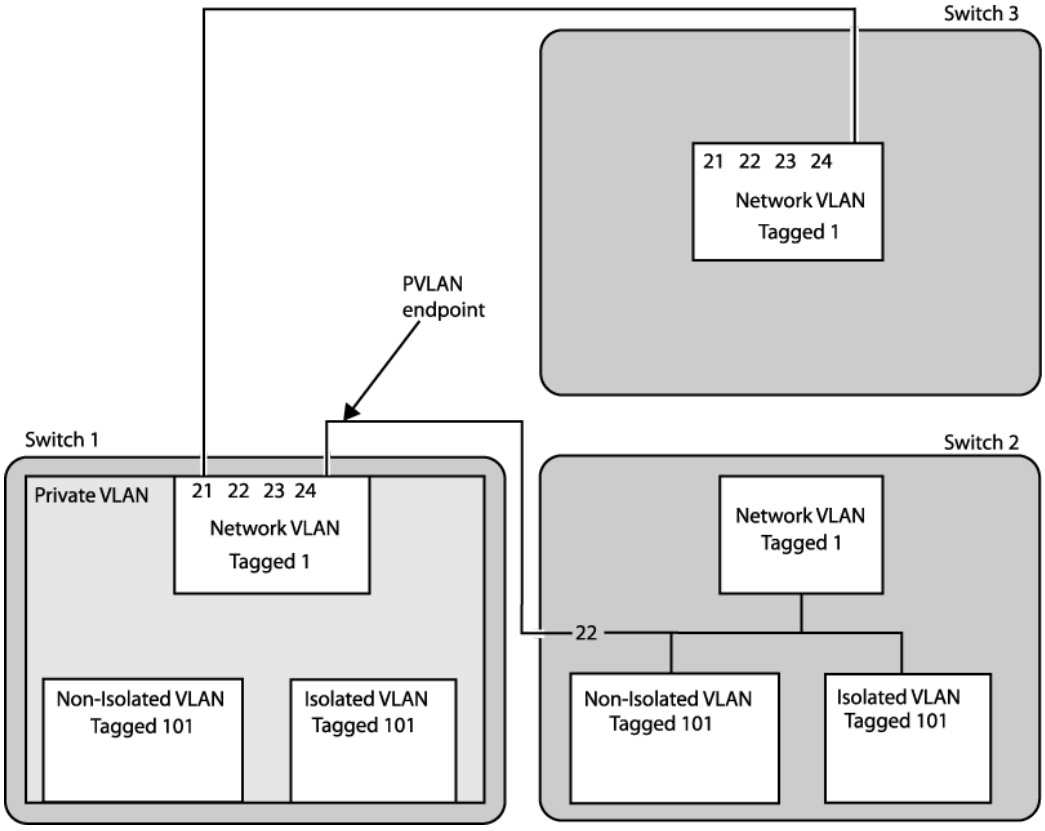


Figure 246: Private VLAN Connections to Switches Outside the PVLAN

In this configuration, Switch 1, Network VLAN Port 21 connects to a Switch 3 port that only supports the Network VLAN. The Network VLAN Port 21 on Switch 1 is configured as “translated,” which translates subscriber VLAN tags to the network VLAN tag for

access to the Network VLAN extension on Switch 3. Switch 3, Port 24 is configured as tagged and only accepts traffic with the Network VLAN Tag. Switch 3 serves as an extension of the Network VLAN and can be used to connect to network devices such as servers or an internet gateway.

Switch 2, port 22 supports the Network, NonIsolated, and Isolated VLANs, but no PVLAN is configured.

Because port 22 supports multiple VLANs that are part of the PVLAN, and because these Switch 2 VLANs are not part of the PVLAN, Switch 1, port 24, must be configured as a PVLAN endpoint, which establishes the PVLAN boundary. Switch 2, port 22, is configured as a regular tagged VLAN port.

For most applications, it would be better to extend the PVLAN to Switch 2 so that the PVLAN features are available to the Switch 2 VLANs.

The configuration of Switch 2 behaves as follows:

- The Switch 2 NonIsolated VLAN ports can communicate with the NonIsolated VLAN ports on Switch 1, but they cannot participate in VLAN translation.
- The Switch 2 Isolated VLAN ports can communicate with other Switch 2 Isolated VLAN ports.
- The Switch 2 Isolated VLAN ports cannot participate in VLAN translation.
- The Switch 2 Isolated VLAN ports can receive broadcast and multicast info for the Isolated VLAN.
- Traffic is allowed from the Switch 1 Isolated VLAN ports to the Switch 2 Isolated VLAN ports.

MAC Address Management in a PVLAN

Each device that connects to a PVLAN must have a unique MAC address within the PVLAN. Each MAC address learned in a PVLAN requires multiple *FDB (forwarding database)* entries. For example, each MAC address learned in a non-isolated subscriber *VLAN* requires two FDB entries, one for the subscriber VLAN and one for the network VLAN. The additional FDB entries for a PVLAN are marked with the P flag in the `show fdb` command display.

The following sections describe the FDB entries created for the PVLAN components and how to estimate the impact of a PVLAN on the FDB table:

- Non-Isolated Subscriber VLAN
- Isolated Subscriber VLAN
- Network VLAN
- Calculating the Total FDB Entries for a PVLAN

Non-Isolated Subscriber VLAN

When a MAC address is learned on a non-isolated subscriber VLAN port, two entries are added to the FDB table:

- MAC address, non-isolated subscriber VLAN tag, and the port number
- MAC address, network VLAN tag, port number, and a special flag for tag translation

The network VLAN entry is used when traffic comes in from the network ports destined for an non-isolated port.

Isolated Subscriber VLAN

When a new MAC address is learned on an isolated subscriber VLAN port, two entries are added to the FDB table:

- MAC address, isolated subscriber VLAN tag, port number, and a flag that indicates that the packet should be dropped
- MAC address, network VLAN tag, port number, and a special flag for tag translation

Ports in the isolated VLAN do not communicate with one another.

If a port in the isolated VLAN sends a packet to another port in the same VLAN that already has an entry in the FDB, that packet is dropped. You can verify the drop packet status of an FDB entry by using the `show fdb` command. The D flag indicates that packets destined for the listed address are dropped.

The network VLAN entry is used when traffic comes in from the network ports destined for an isolated port.

Network VLAN

When a new MAC address is learned on a network VLAN port, the following entry is added to the FDB table: MAC address, network VLAN tag, and port number.

For every subscriber VLAN belonging to this PVLAN, the following entry is added to the FDB table: MAC address, subscriber VLAN tag, and port number

Calculating the Total FDB Entries for a PVLAN

The following formula can be used to estimate the maximum number of FDB entries for a PVLAN:

FDBtotal = [(MACnon-iso + MACiso) * 2 + (MACnetwork * (VLANnon-iso + VLANiso + 1))]

The formula components are as follows:

- MACnon-iso = number of MAC addresses learned on all the non-isolated subscriber VLANs
- MACiso = number of MAC addresses learned on all the isolated subscriber VLANs
- MACnetwork = number of MAC addresses learned on the network VLAN
- VLANnon-iso = number of non-isolated subscriber VLANs
- VLANiso = number of isolated subscriber VLANs

Note
The formula above estimates the worst-case scenario for the maximum number of FDB entries for a single PVLAN. If the switch supports additional PVLANs, apply the formula to each PVLAN and add the totals for all PVLANs. If the switch also support standard VLANs, there will also be FDB entries for the standard VLANs.

Layer 3 Communications

For PVLANs, the default switch configuration controls Layer 3 communications exactly as communications are controlled in Layer 2.

For example, Layer 3 communications is enabled between ports in a non-isolated subscriber *VLAN*, and disabled between ports in an isolated subscriber VLAN. Ports in a non-isolated subscriber VLAN cannot communicate with ports in other non-isolated subscriber VLANs.

You can enable Layer 3 communications between all ports in a PVLAN. For more information, see Managing Layer 3 Communications in a PVLAN on page 2242.

PVLAN Limitations

The Private *VLAN* feature has the following limitations:

- Requires more *FDB* entries than a standard VLAN.
- VLAN tag duplication is not allowed.
- VLAN name duplication is not allowed.
- Each MAC address learned in a PVLAN must be unique. A MAC address cannot exist in two or more VLANs that belong to the same PVLAN.
- MVR cannot be configured on PVLANs.
- A VMAN cannot be added to a PVLAN.
- A PBB network (BVLAN) cannot be added to a PVLAN.
- *EAPS* control VLANs cannot be either subscriber or network VLANs.
- For PVLAN with *STP (Spanning Tree Protocol)* implementation, irrespective of port translation configuration in the Network VLAN, it is recommended to add both the Network VLAN and all subscriber VLANs to the STP.
- For PVLAN with EAPS implementation, irrespective of port translation configuration in the Network VLAN, it is recommended to add both the Network VLAN and all subscriber VLANs to the EAPS ring.
- *ESRP (Extreme Standby Router Protocol)* can only be configured on network VLAN ports (and not on subscriber VLAN ports). To support ESRP on the network VLAN, you must add all of the VLANs in the PVLAN to ESRP.
- There is no *NetLogin* support to add ports as translate to the network VLAN, but the rest of NetLogin and the PVLAN features do not conflict.
- *IGMP (Internet Group Management Protocol)* snooping is performed across the entire PVLAN, spanning all the subscriber VLANs, following the PVLAN rules. For VLANs that are not part of a PVLAN, IGMP snooping operates as normal.
- PVLAN and VPLS are not supported on the same VLAN.
- When two switches are part of the same PVLAN, unicast and multicast traffic require a tagged trunk between them that preserves tags (no tag translation).
- Subscriber VLANs in a PVLAN cannot exchange multicast data with VLANs outside the PVLAN and with other PVLANs. However, the network VLAN can exchange multicast data with VLANs outside the PVLAN and with network VLANs in other PVLANs.

- PVLAN does not support IPv6.
- DHCP does not work on subscriber VLANs because an IP address can not be assigned to the subscriber VLAN.

Note
A maximum of 80% of 4K VLANs can be added to a PVLAN. Adding more VLANS will display the following log error:
`<Erro:HAL.VLAN.Error>Slot-<slot>: Failed to add egress vlan translation entry on port <port> due to "Table full".`

If two or more member VLANs have overlapping ports (where the same ports are assigned to both VLANs), each additional VLAN member with overlapping ports must have a dedicated loopback port. To state it another way, one of the VLAN members with overlapping ports does not require a dedicated loopback port, and the rest of the VLAN members do require a single, dedicated loopback port within each member VLAN.

Note
There is a limit to the number of unique source MAC addresses on the network VLAN of a PVLAN that the switch can manage. It is advised not to exceed the value shown in the item "FDB (maximum L2 entries)" in the Supported Limits table of the *ExtremeXOS Release Notes*.

Configuring PVLANs

The following section describes how to configure a private *VLAN*.

Creating PVLANs

To create a *VLAN*, you need to do the following:

1. Create the PVLAN.
2. Add one VLAN to the PVLAN as a network VLAN.
3. Add VLANs to the PVLAN as subscriber VLANs.

- To create a PVLAN, use the following command:

  `create private-vlan` *name* {**vr** *vr_name*}
- To add a network VLAN to the PVLAN, create and configure a tagged VLAN, and then use the following command to add that network VLAN:

  `configure private-vlan` *name* **add network** *vlan_name*
- To add a subscriber VLAN to the PVLAN, create and configure a tagged VLAN, and then use the following command to add that subscriber VLAN:

  `configure private-vlan` *name* **add subscriber** *vlan_name* {**non-isolated**} {**loopback-port** *port*}

  By default, this command adds an isolated subscriber VLAN. To create a non-isolated subscriber VLAN, you must include the non-isolated option.

Configuring Network VLAN Ports for VLAN Translation

When subscriber *VLAN* traffic exits a network VLAN port, it can be untagged, tagged (with the subscriber VLAN tag), or translated (to the network VLAN tag).

Note
All traffic that exits a subscriber VLAN port uses the subscriber VLAN tag, unless the port is configured as untagged. There is no need to configure VLAN translation (from network to subscriber VLAN tag) on subscriber VLAN ports.

1. To configure network VLAN ports for VLAN translation, use the following command and specify the network VLAN and port numbers:

   ```
   configure {vlan} vlan_name | vlan_list add ports port_list private-vlan translated
   ```

2. If you want to later reconfigure a port that is configured for VLAN translation so that it does not translate tags, use the following command and specify either the tagged or the untagged option:

   ```
   configure {vlan} vlan_name | vlan_list add ports [port_list | all] {tagged | untagged} {{stpd} stpd_name} {dot1d | emistp | pvst-plus}}
   ```

Configuring Non-Isolated Subscriber VLAN Ports

The process for configuring non-isolated *VLAN* ports requires two tasks:

- Add a VLAN to the PVLAN as a non-isolated subscriber VLAN.
- Assign ports to the non-isolated subscriber VLAN.

These tasks can be completed in any order, but they must both be completed before a port can participate in a PVLAN. When configuration is complete, all egress traffic from the port is translated to the VLAN tag for that non-isolated VLAN (unless the port is configured as untagged).

Note
To configure VLAN translation for network VLAN ports, see Configuring Network VLAN Ports for VLAN Translation on page 2240.

- To add a non-isolated subscriber VLAN to the PVLAN, use the following command:

  ```
  configure private-vlan name add subscriber vlan_name non-isolated
  ```

- To add ports to a non-isolated VLAN (before or after it is added to the PVLAN), use the following command:

  ```
  configure {vlan} vlan_name | vlan_list add ports [port_list | all] {tagged | untagged} {{stpd} stpd_name} {dot1d | emistp | pvst-plus}}
  ```

If you specify the tagged option, egress traffic uses the non-isolated VLAN tag, regardless of the network translation configuration on any network port with which these ports communicate. Egress traffic from a non-isolated VLAN port never carries the network VLAN tag.

Configuring Isolated Subscriber VLAN Ports

When a port is successfully added to an isolated *VLAN*, the port is isolated from other ports in the same VLAN, and all egress traffic from the port is translated to the VLAN tag for that VLAN (unless the port is configured as untagged).

Note
To configure VLAN translation for network VLAN ports, see Configuring Network VLAN Ports for VLAN Translation on page 2240.

The process for configuring ports for VLAN isolation requires two tasks:

- Add a VLAN to the PVLAN as an isolated subscriber VLAN.
- Assign ports to the isolated subscriber VLAN.

These tasks can be completed in any order, but they must both be completed before a port can participate in an isolated VLAN.

- To add an isolated subscriber VLAN to the PVLAN, use the following command:
  `configure private-vlan` *`name`* **`add subscriber`** *`vlan_name`*
- To add ports to an isolated VLAN (before or after it is added to the PVLAN), use the following command:
  `configure {`**`vlan`**`}` *`vlan_name`* | *`vlan_list`* **`add ports`** `[`*`port_list`* | **`all`**`]`
  `{`**`tagged`** | **`untagged`**`} {{`**`stpd`**`}` *`stpd_name`*`} {`**`dot1d`** | **`emistp`** | **`pvst-plus`**`}}`

If you specify the tagged option, egress traffic uses the isolated VLAN tag, regardless of the network translation configuration on any network port with which these ports communicate. Egress traffic from an isolated VLAN port never carries the network VLAN tag.

Configuring a PVLAN on Multiple Switches

To create a PVLAN that runs on multiple switches, you must configure the PVLAN on each switch and set up a connection between the network VLANs on each switch. The ports at each end of the connection must be configured as tagged ports that do not perform tag translation.

To configure these types of ports, use the following command:

`configure {`**`vlan`**`}` *`vlan_name`* **`add ports`** *`port_list`* **`tagged`**

Configuring a Network or Subscriber VLAN Extension to Another Switch

You can extend a network or subscriber *VLAN* to another switch without configuring a PVLAN on that switch. This configuration is introduced in Extending Network and Subscriber VLANs to Other Switches on page 2235.

- To configure the port on the switch that is outside of the PVLAN, use the following command:
  `configure {`**`vlan`**`}` *`vlan_name`* **`add ports`** *`port_list`* **`tagged`**

Adding a Loopback Port to a Subscriber VLAN

ExtremeSwitching series switches, whether or not included in a SummitStack, require a loopback port for certain configurations. If two or more subscriber *VLAN*s have

overlapping ports (where the same ports are assigned to both VLANs), each of the subscriber VLANs with overlapping ports must have a dedicated loopback port.

The loopback port can be added when the subscriber VLAN is added to the PVLAN.

If you need to add a loopback port to an existing subscriber VLAN, use the following command:

```
configure {vlan} vlan_name vlan-translation add loopback-port port
```

Managing Layer 3 Communications in a PVLAN

The default configuration for Layer 3 PVLAN communications is described in Layer 3 Communications.

To enable Layer 3 communications between all ports in a PVLAN, use the following command:

```
configure iparp add proxy [ipNetmask | ip_addr {mask}] {vr vr_name} {mac | vrrp} {always}
```

Specify the IP address or subnet specified for the network *VLAN* in the PVLAN. Use the **always** option to ensure that the switch will reply to ARP requests, regardless of the VLAN from which it originated.

Delete PVLANs

To delete an existing PVLAN, use the command:

```
delete private-vlan name
```

Remove a VLAN from a PVLAN

When you remove a *VLAN* from a PVLAN, you remove the association between a VLAN and the PVLAN. Both the VLAN and PVLAN exist after the removal.

To remove a network or subscriber VLAN from a PVLAN, use the following command:

```
configure private-vlan name delete [network | subscriber] vlan_name
```

Deleting a Loopback Port from a Subscriber VLAN

To delete a loopback port from a subscriber *VLAN*, use the command:

```
configure {vlan} vlan_name vlan-translation delete loopback-port
```

Displaying PVLAN Information

This section describes how to display private *VLAN* information.

Displaying Information for all PVLANs

To display information on all the PVLANs configured on a switch, use the command:

```
show private-vlan
```

Displaying Information for a Specific PVLAN

To display information about a single PVLANs, use the command:

```
show {private-vlan} name
```

Displaying Information for a Network or Subscriber VLAN

To display information about a network or subscriber *VLAN*, use the command:

```
show vlan {virtual-router vr-name}
```

The following flags provide PVLAN specific information:

s flat

Identifies a network VLAN port that the system added to a subscriber VLAN. All subscriber VLANs contain network VLAN ports that are marked with the s flag.

L flag

Identifies a subscriber VLAN port that is configured as a loopback port.

t flag

Identifies a tagged network VLAN port on which tag translation is enabled. The t flag only appears in the show vlan display for network VLANs.

e flag

Identifies a network VLAN port that is configured as an endpoint. The e flag only appears in the show vlan display for network VLANs.

Displaying PVLAN FDB Entries

To view all *FDB* entries including those created for a PVLAN, use the command:

```
show fdb {blackhole {netlogin [all | mac-based-vlans]} | netlogin [all
| mac-based-vlans] | permanent {netlogin [all | mac-based-vlans]} |
mac_addr {netlogin [all | mac-based-vlans]} | ports port_list {netlogin
[all | mac-based-vlans]} | vlan vlan_name | vlan_list {netlogin [all |
mac-based-vlans]} | {{vpls} {vpls_name}}}
```

The P flag marks additional FDB entries for PVLANs.

PVLAN Configuration Example 1

Figure 247 on page 2244 shows a PVLAN configuration example for a medical research lab.

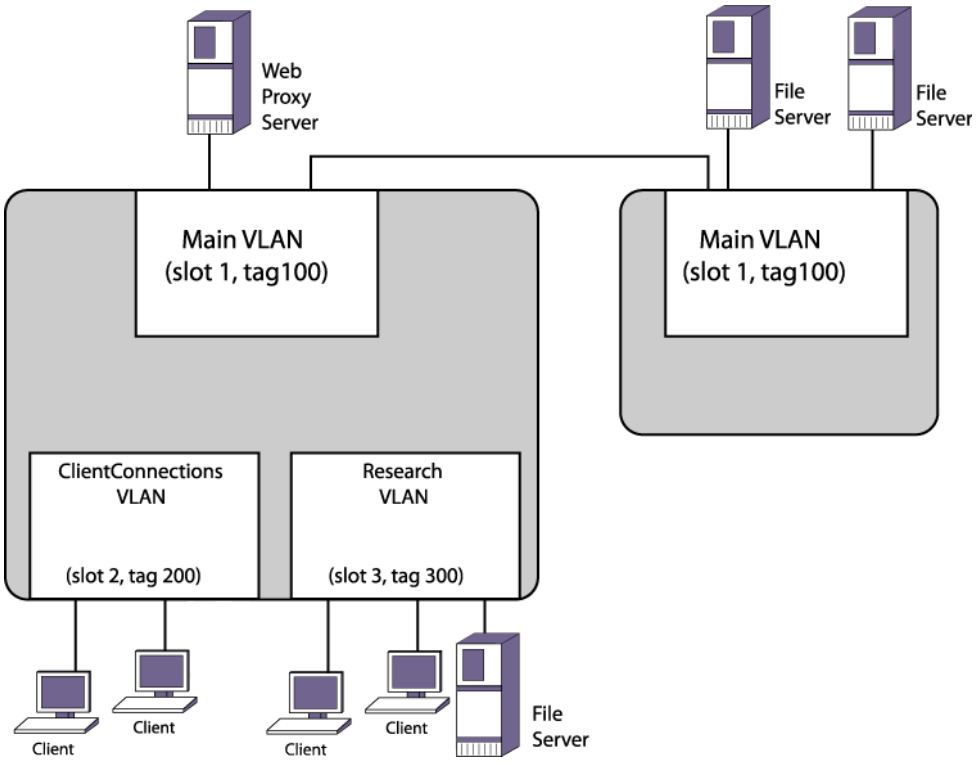


Figure 247: PVLAN Configuration Example 1

The medical research lab hosts lots of visiting clients. Each client has their own room, and the lab wants to grant them access to the internet through a local web proxy server but prevent them from accessing other visiting clients. There is a lab in the building where many research workstations are located. Workstations within the lab require access to other lab workstations, the internet, and file servers that are connected to a switch in another building. Visiting clients should not have access to the Research *VLAN* devices or the file servers on the remote switch.

The PVLAN in Figure 247 contains the following PVLAN components:

- Network VLAN named Main, which provides internet access through the proxy web server and access to file servers on the remote switch.
- Isolated subscriber VLAN named ClientConnections, which provides internet access for visiting clients and isolation from other visiting clients, the Research VLAN devices, and the remote file servers.
- Non-isolated subscriber VLAN named Research, which provides internet access and enables communications between Research VLAN devices and the remote file servers.

1. The first configuration step is to create and configure the VLANs on the local switch:

```
create vlan Main
configure vlan Main add port 1:*
configure vlan Main tag 100
create vlan ClientConnections
configure vlan ClientConnections add port 2:*
configure vlan ClientConnections tag 200
create vlan Research
configure vlan Research add port 3:*
configure vlan Research tag 300
```

2. The remote switch VLAN is configured as follows:

```
create vlan Main
configure vlan Main add port 1:*
configure vlan Main tag 100
```

3. The next step is to create the PVLAN on the local switch and configure each of the component VLANs for the proper role:

```
create private-vlan MedPrivate
configure private-vlan "MedPrivate" add network "Main"
configure private-vlan "MedPrivate" add subscriber "ClientConnections"
configure private-vlan "MedPrivate" add subscriber "Research" non-isolated
```

4. The final step is to configure VLAN translation on the local switch so that Research VLAN workstations can connect to the file servers on the remote switch:

```
configure Main add ports 1:1 private-vlan translated
```

5. To view the completed configuration, enter the `show private-vlan` command as follows:

```
show private-vlan
-------------------------------------------------------------------------------------
Name            VID  Protocol Addr        Flags                Proto  Ports  Virtual
Active router
/Total
-------------------------------------------------------------------------------------
MedPrivate                                                                 VR-Default
Network VLAN:
-main          100  ------------------------------------    ANY    2 /48  VR-Default
Non-Isolated Subscriber VLAN:
-Research      300  ------------------------------------    ANY    2 /96  VR-Default
Isolated Subscriber VLAN:
-ClientConnections 200  ---------------------------------    ANY    2 /52  VR-Default
```

PVLAN Configuration Example 2

Figure 248 shows a PVLAN configuration example for a motel.

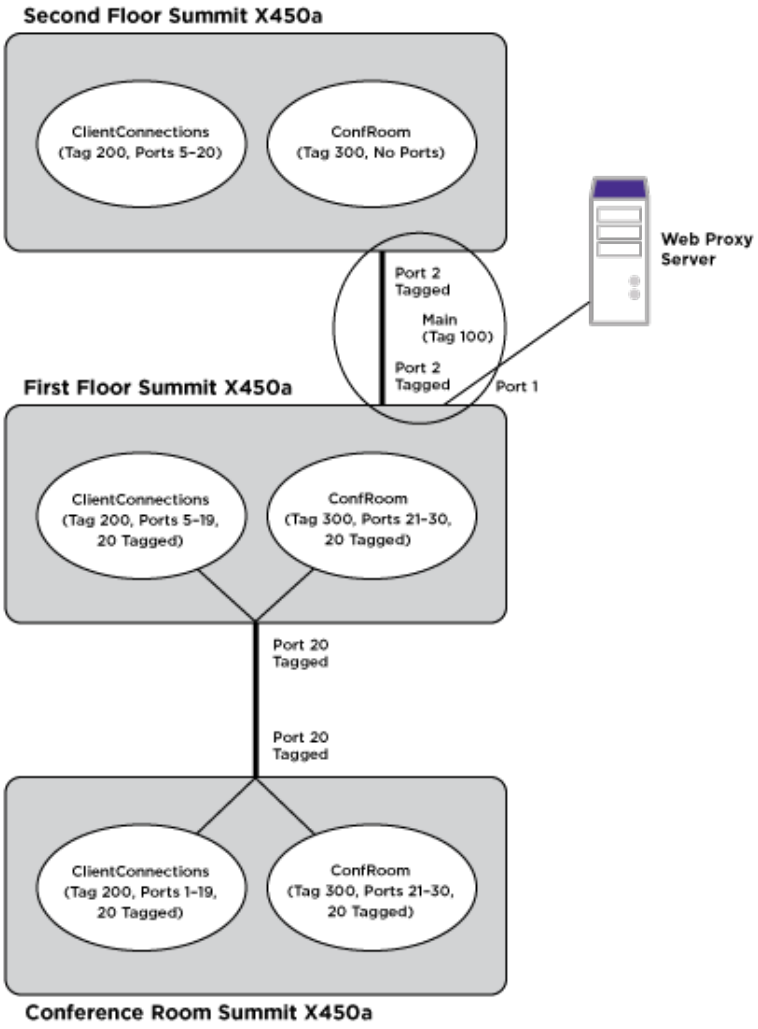


Figure 248: PVLAN Configuration Example 2

This motel has guest rooms, a conference room, and their web proxy server on the first floor, and guest rooms on the second floor. The motel has three ExtremeSwitching switches. There is one on the first floor in a closet, one on the first floor in the conference room, and one on the second floor.

The PVLAN in Figure 248 contains the following PVLAN components:

- A *VLAN* called Main that contains the web proxy server.
- A VLAN called ConfRoom that contains the ports for the conference room connections.
- A VLAN called ClientConnections that contains client PC connections for the guest rooms.

The goals for the motel network are as follows:

- Provide internet access for the ConfRoom and ClientConnections VLANs through the web proxy server.
- Prevent communications between the ConfRoom and ClientConnections VLANs.
- Enable communications between clients on the ClientConnections VLAN only within the conference room.
- Enable communications between devices on the ConfRoom VLAN.
- Prevent communications between the PCs in the ClientConnections VLAN that are not in the conference room.

Notice the following in this example:

- The ExtremeSwitching switches in the first floor closet and on the second floor contain the Main VLAN with a tag of 100. This VLAN is connected via a tagged port between the first and second floor switches.
- The switch in the conference room does not contain the Main VLAN and cannot be a PVLAN member.
- All of the switches have the ClientConnections VLAN, and it uses VLAN tag 200.
- All of the switches have the ConfRoom VLAN, and it uses VLAN tag 300.
- The Conference Room switch connects to the rest of the network through a tagged connection to the switch in the first floor closet.
- Because the switch in the first floor closet is a PVLAN member and uses the same port to support two subscriber VLANs, a loopback port is required in all subscriber VLANs, except the first configured subscriber VLAN (this applies to all switches).

Note
The following examples contain comments that follow the CLI comment character (#). All text that follows this character is ignored by the switch and can be omitted from the switch configuration.

The following commands configure the switch in the first floor closet:

```
# Create and configure the VLANs.
create vlan Main
configure vlan Main add port 1
configure vlan Main tag 100
configure vlan Main add port 2 tagged
create vlan ClientConnections
configure vlan ClientConnections tag 200
configure vlan ClientConnections add port 5-19
configure vlan ClientConnections add port 20 tagged
create vlan ConfRoom
configure vlan ConfRoom tag 300
configure vlan ConfRoom add port 21-30
configure vlan ConfRoom add port 20 tagged

# Create and configure the PVLAN named Motel.
create private-vlan Motel
configure private-vlan Motel add network Main
configure private-vlan Motel add subscriber ClientConnections # isolated subscriber VLAN
configure private-vlan "Motel" add subscriber "ConfRoom" non-isolated loopback-port 30
configure private-vlan Motel add subscriber ConfRoom non-isolated
# If you omit the loopback-port command, the above command produces the following error
message:
```

```
# Cannot add subscriber because another subscriber vlan is already present on the same
port, assign a loopback port when adding the subscriber vlan to the private vlan
```

```
# show vlan "ConfRoom"
VLAN Interface with name ConfRoom created by user
Admin State:    Enabled         Tagging:        802.1Q Tag 300
Virtual router: VR-Default
IPv6:           None
STPD:           None
Protocol:       Match all unfiltered protocols
Loopback:       Disabled
NetLogin:       Disabled
QosProfile:     None configured
Egress Rate Limit Designated Port: None configured
Private-VLAN Name:              Motel
VLAN Type in Private-VLAN:      Non-Isolated Subscriber
Ports:   13.      (Number of active ports=1)
Untag:      21,     22,     23,     24,     25,     26,     27,
28,     29
Tag:        1s,     2s,     20,   *30L
Flags:    (*) Active, (!) Disabled, (g) Load Sharing port
(b) Port blocked on the vlan, (m) Mac-Based port
(a) Egress traffic allowed for NetLogin
(u) Egress traffic unallowed for NetLogin
(t) Translate VLAN tag for Private-VLAN
(s) Private-VLAN System Port, (L) Loopback port
(x) VMAN Tag Translated port
(F) Dynamically added by Fabric Attach
(G) Multi-switch LAG Group port
(H) Dynamically added by MVRP
(I) Dynamically added by IDM
(U) Dynamically added uplink port
(V) Dynamically added by VM Tracking

# Note that the loopback port is flagged with an "L" and listed as a tagged port, and the
network VLAN ports are flagged with an "s" and listed as tagged ports.
```

The following commands configure the switch on the second floor:

```
# create and configure the VLANs
create vlan Main
configure vlan Main tag 100
configure vlan Main add port 2 tagged
create vlan ClientConnections
configure vlan ClientConnections tag 200
configure vlan ClientConnections add port 5-20
create vlan ConfRoom
configure vlan ConfRoom tag 300
# Create and configure the PVLAN named Motel.
create private-vlan Motel
configure private-vlan Motel add network Main
configure private-vlan Motel add subscriber ClientConnections # isolated subscriber VLAN
configure private-vlan Motel add subscriber ConfRoom non-isolated
```

The following commands configure the switch in the conference room:

```
# create and configure the VLANs
create vlan ClientConnections
configure vlan ClientConnections tag 200
configure vlan ClientConnections add port 1-19
configure vlan ClientConnections add port 20 tag
create vlan ConfRoom
configure vlan ConfRoom tag 300
configure vlan ConfRoom add port 21-30
```

```
configure vlan ConfRoom add port 20 tag
# The VLANs operate as extensions of the VLANs on the Summit in the first floor closet.
There is no PVLAN configuration on this switch.
```

VLAN Translation

The *VLAN* translation feature described in this section provides the same VLAN translation functionality that is provided for PVLANs. This is described in VLAN Translation in a PVLAN on page 2230.

Note

The VLAN translation feature is supported only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

Figure 249 shows how VLAN translation is configured in the switch.

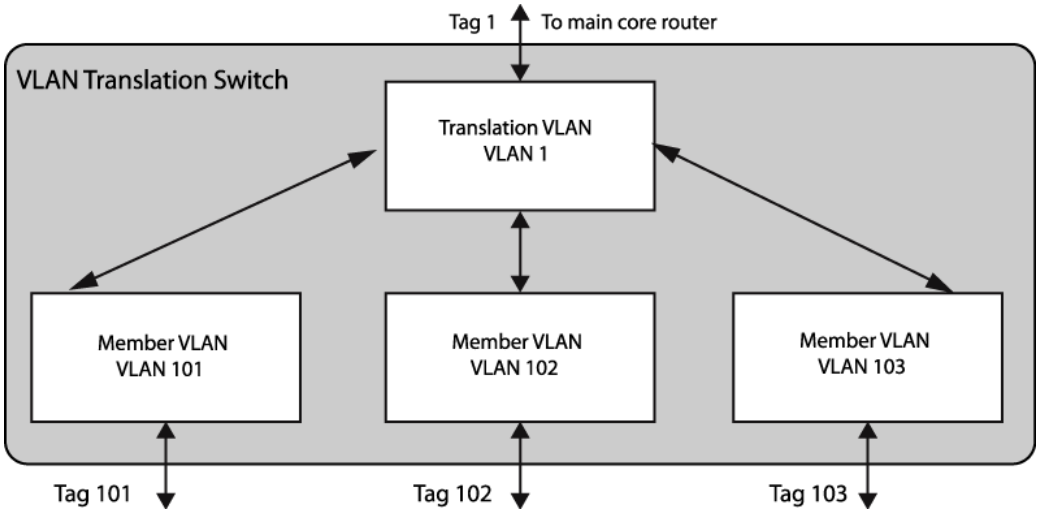


Figure 249: VLAN Translation Switch Configuration

VLAN1 is configured as a translation VLAN . The translation VLAN is equivalent to the network VLAN in the PVLAN implementation of VLAN translation.

VLANs 101, 102, and 103 are configured as member VLANs of translation VLAN1. The member VLANs are equivalent to the non-isolated subscriber VLANs in the PVLAN implementation of VLAN translation.

This configuration enables tag translation between the translation VLAN and the member VLANs. All member VLANs can communicate through the translation VLAN, but they cannot communicate through Layer 2 with each other.

VLAN Translation Behavior

You should be aware of the behavior of unicast, broadcast, and multicast traffic when using *VLAN* translation.

Unicast Traffic

Traffic on the member *VLAN*s can be either tagged or untagged.

Traffic is switched locally between client devices on the same member VLAN as normal. Traffic cannot be switched between clients on separate member VLANs. Traffic from any member VLAN destined for the translation VLAN is switched and the VLAN tag is translated appropriately. Traffic from the translation VLAN destined for any member VLAN is switched and the VLAN tag is translated.

Broadcast Behavior

Broadcast traffic generated on a member *VLAN* is replicated in every other active port of that VLAN as normal.

In addition, the member VLAN traffic is replicated to every active port in the translation VLAN and the VLAN tag is translated appropriately. Broadcast traffic generated on the translation VLAN is replicated to every other active port in this VLAN as usual. The caveat in this scenario is that this traffic is also replicated to every active port in every member VLAN, with VLAN tag translation. In effect, the broadcast traffic from the translation VLAN leaks onto all member VLANs.

Multicast Behavior

IGMP snooping can be enabled on member and translation *VLAN*s so that multicast traffic can be monitored within the network.

IGMP snooping software examines all IGMP control traffic that enters the switch. IGMP control traffic received on a VLAN translation port is forwarded by the CPU to all other ports in the translation group. Software VLAN translation is performed on the packets which cross the translation boundary between member and translation VLANs. The snooping software detects ports joining and leaving multicast streams. When a VLAN translation port joins a multicast group, an *FDB* entry is installed only on receiving a data miss for that group. The FDB entry is added for the requested multicast address and contains a multicast PTAG. When a VLAN translation port leaves a multicast group, the port is removed from the multicast list. The last VLAN translation port to leave a multicast group causes the multicast FDB entry to be removed.

VLAN Translation Limitations

The *VLAN* translation feature has the following limitations:

- Requires more *FDB* entries than a standard VLAN.
- VLAN tag duplication is not allowed.
- VLAN name duplication is not allowed.
- Each MAC address learned in the translation and member VLANs must be unique. A MAC address cannot exist in two or more VLANs that belong to the same VLAN translation domain.
- MVR cannot be configured on translation and member VLANs.
- A VMAN cannot be added to translation and member VLANs.
- A PBB network (BVLAN) cannot be added to translation and member VLANs.

- *EAPS* control VLANs cannot be either translation or member VLANs.
- EAPS can only be configured on translation VLAN ports (and not on member VLAN ports). To support EAPS on the network VLAN, you must add all of the translation and member VLANs to the EAPS ring.
- *STP* can only be configured on translation VLAN ports (and not on member VLAN ports). To support STP on the translation VLAN, you must add the translation VLAN and all of the member VLANs to STP.
- *ESRP* can only be configured on translation VLAN ports (and not on member VLAN ports). To support ESRP on the network VLAN, you must add the translation VLAN and all of the member VLANs to ESRP.
- There is no *NetLogin* support to add ports as translate to the translation VLAN, but the rest of NetLogin and the PVLAN feature do not conflict.
- *IGMP* snooping is performed across the entire VLAN translation domain, spanning all the member VLANs. For VLANs that are not part of a VLAN translation domain, IGMP snooping operates as normal.
- VLAN translation and VPLS are not supported on the same VLAN.
- Member VLANs in a VLAN translation domain cannot exchange multicast data with VLANs outside the VLAN translation domain. However, the translation VLAN can exchange multicast data with VLANs outside the VLAN translation domain and with translation VLANs in other VLAN translation domains.

Interfaces

Use the following information for selecting and configuring *VLAN* translation interfaces:

- A single physical port can be added to multiple member VLANs, using different VLAN tags.
- Member VLANs and translation VLANs can include both tagged and untagged ports.

Configuring Translation VLANs

To create a translation *VLAN*, do the following:

1. Create the VLAN that will become the translation VLAN.
2. Add a tag and ports to the prospective translation VLAN.
3. Add member VLANs to the prospective translation VLAN.

A prospective translation VLAN becomes a translation VLAN when the first member VLAN is added to it.

- To add a member VLAN to a translation VLAN, use the following command:
  `configure {vlan} vlan_name vlan-translation add member-vlan member_vlan_name {loopback-port port}`
- To delete a member VLAN from a translation VLAN, use the following command:
  `configure {vlan} vlan_name vlan-translation delete member-vlan [member_vlan_name | all]`
- To view the translation VLAN participation status of a VLAN, use the following command:
  `show vlan {virtual-router vr-name}`

Displaying Translation VLAN Information

This section describes how to display translation *VLAN* information.

Displaying Information for a Translation or Member VLAN

To display information about a translation or member *VLAN*, use the command:

```
show vlan {virtual-router vr-name}
```

Displaying Translation VLAN FDB Entries

To view all *FDB* entries including those created for a translation *VLAN*, use the command:

```
show fdb {blackhole {netlogin [all | mac-based-vlans]} | netlogin [all
| mac-based-vlans] | permanent {netlogin [all | mac-based-vlans]} |
mac_addr {netlogin [all | mac-based-vlans]} | ports port_list {netlogin
[all | mac-based-vlans]} | vlan vlan_name | vlan_list {netlogin [all |
mac-based-vlans]} | {{vpls} {vpls_name}}}
```

The T flag marks additional FDB entries for translation VLANs.

VLAN Translation Configuration Examples

The following configuration examples show *VLAN* translation used in three scenarios:

- Basic VLAN Translation on page 2252
- VLAN Translation with ESRP Redundancy on page 2253
- VLAN Translation with STP Redundancy on page 2254

Basic VLAN Translation

The example in Figure 250configures a basic *VLAN* translation network. This network provides VLAN translation between four member VLANs and a single translation VLAN.

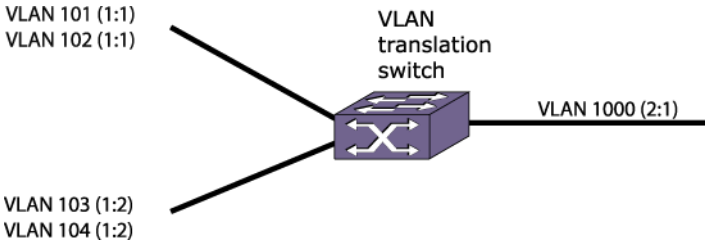


Figure 250: VLAN Translation Configuration Example

The following configuration commands create the member VLANs:

```
create vlan v101
configure v101 tag 101
configure v101 add ports 1:1 tagged
create vlan v102
configure v102 tag 102
configure v102 add ports 1:1 tagged
create vlan v103
configure v103 tag 103
configure v103 add ports 1:2 tagged
```

```
create vlan v104
configure v104 tag 104
configure v104 add ports 1:2 tagged
```

The following configuration commands create the translation VLAN and enable VLAN translation:

```
create vlan v1000
configure v1000 tag 1000
configure v1000 add ports 2:1 tagged
configure v1000 vlan-translation add member-vlan v101
configure v1000 vlan-translation add member-vlan v102 loopback-port 1:23
configure v1000 vlan-translation add member-vlan v103
configure v1000 vlan-translation add member-vlan v104 loopback-port 1:24
```

VLAN Translation with ESRP Redundancy

The example in Figure 251 configures a *VLAN* translation network with *ESRP* redundancy.

The SW2 and SW3 VLAN translation switches are protected by an ESRP control VLAN. The master ESRP switch performs the translation and provides the connectivity to the backbone. If a failure occurs, the slave ESRP switch takes over and begins performing the translation.

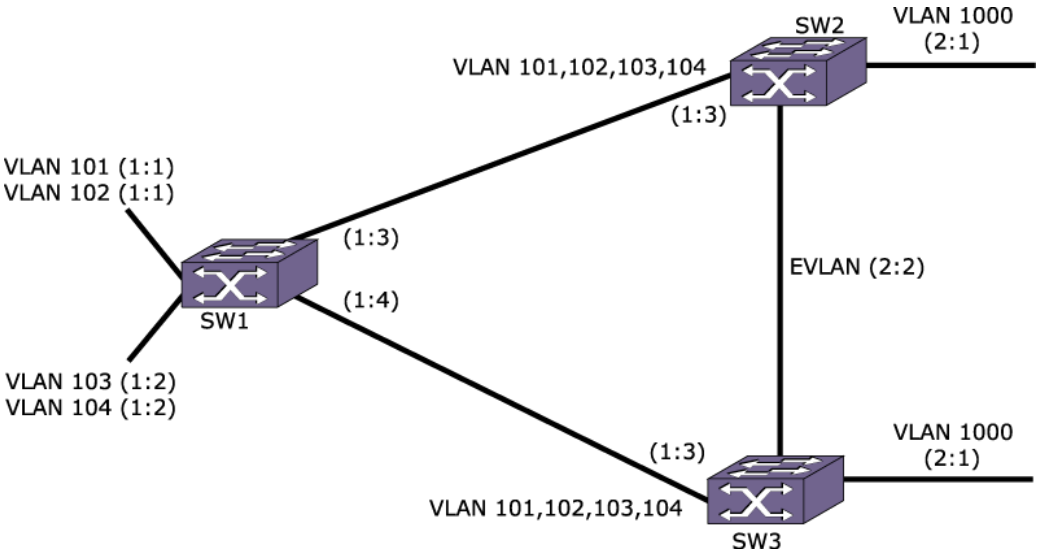


Figure 251: ESRP Redundancy Configuration Example

The following configuration commands create the member VLANs on SW1:

```
create vlan v101
configure v101 tag 101
configure v101 add ports 1:1 tagged
configure v101 add ports 1:3 tagged
configure v101 add ports 1:4 tagged
create vlan v102
configure v102 tag 102
configure v102 add ports 1:1 tagged
configure v102 add ports 1:3 tagged
configure v102 add ports 1:4 tagged
create vlan v103
configure v103 tag 103
configure v103 add ports 1:2 tagged
```

```
configure v103 add ports 1:3 tagged
configure v103 add ports 1:4 tagged
create vlan v104
configure v104 tag 104
configure v104 add ports 1:2 tagged
configure v104 add ports 1:3 tagged
configure v104 add ports 1:4 tagged
```

The configuration for SW2 and SW3 is identical for this example.

The following configuration commands create the member VLANs on SW2:

```
create vlan v101
configure v101 tag 101
configure v101 add ports 1:3 tagged
create vlan v102
configure v102 tag 102
configure v102 add ports 1:3 tagged
create vlan v103
configure v103 tag 103
configure v103 add ports 1:3 tagged
create vlan v104
configure v104 tag 104
configure v104 add ports 1:3 tagged
```

This set of configuration commands creates the translation VLANs and enables VLAN translation on SW2:

```
create vlan v1000
configure v1000 tag 1000
configure v1000 add ports 2:1 tagged
configure v1000 vlan-translation add member-vlan v101
configure v1000 vlan-translation add member-vlan v102
configure v1000 vlan-translation add member-vlan v103
configure v1000 vlan-translation add member-vlan v104
```

The final set of configuration commands creates the ESRP control VLAN and enables ESRP protection on the translation VLAN for SW2:

```
create vlan evlan
configure evlan add ports 2:2
enable esrp evlan
configure evlan add domain-member v1000
```

The following configuration commands create the translation VLAN and enable VLAN translation onVLANs that have overlapping ports:

```
configure v1000 vlan-translation add member-vlan v102 loopback-port 1:22
configure v1000 vlan-translation add member-vlan v103 loopback-port 1:23
configure v1000 vlan-translation add member-vlan v104 loopback-port 1:24
```

VLAN Translation with STP Redundancy

The example in Figure 252 configures a *VLAN* translation network with redundant paths protected by *STP*.

Parallel paths exist from the member VLAN portion of the network to the translation switch. STP ensures that the main path for this traffic is active and the secondary path is blocked. If a failure occurs in the main path, the secondary paths are enabled.

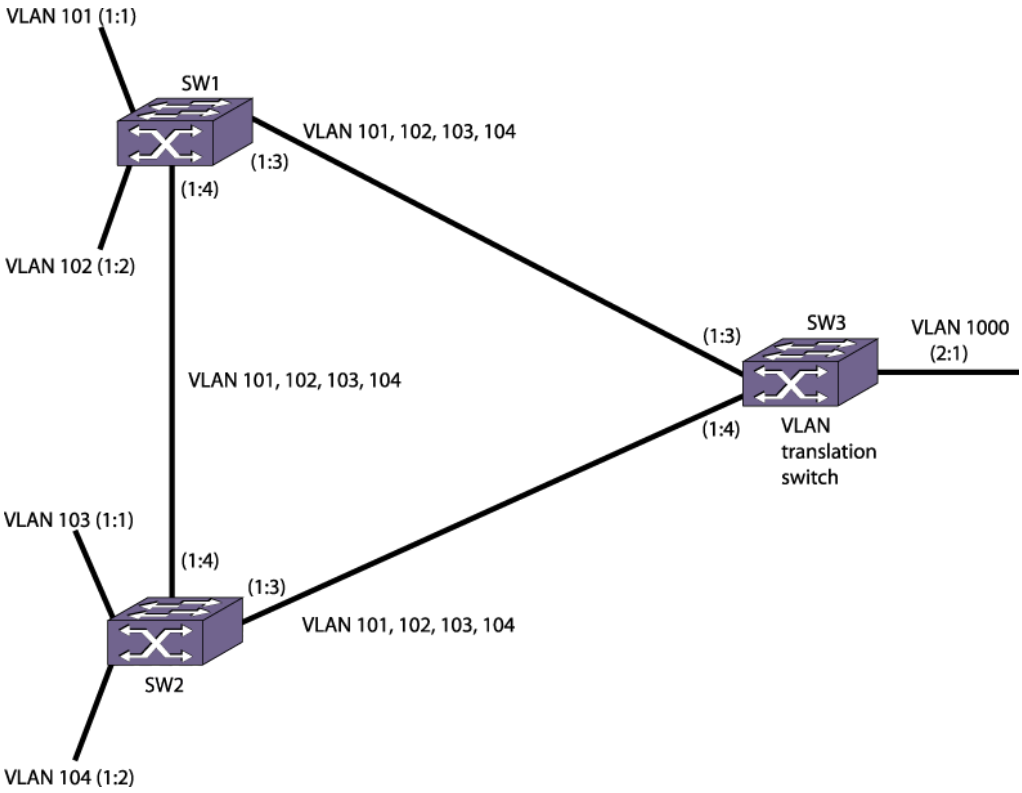


Figure 252: STP Redundancy Configuration Example

The following configuration commands create the member VLANs and enable STP on SW1:

```
create vlan v101
configure v101 tag 101
configure v101 add ports 1:1 tagged
configure v101 add ports 1:3 tagged
configure v101 add ports 1:4 tagged
create vlan v102
configure v102 tag 102
configure v102 add ports 1:2 tagged
configure v102 add ports 1:3 tagged
configure v102 add ports 1:4 tagged
create vlan v103
configure v103 tag 103
configure v103 add ports 1:3 tagged
configure v103 add ports 1:4 tagged
create vlan v104
configure v104 tag 104
configure v104 add ports 1:3 tagged
configure v104 add ports 1:4 tagged
create stpd stp1
configure stp1 tag 101
configure stp1 add vlan v101
configure stp1 add vlan v102
configure stp1 add vlan v103
configure stp1 add vlan v104
enable stpd stp1
```

These configuration commands create the member VLANs and enable STP on SW2:

```
create vlan v103
configure v103 tag 103
configure v103 add ports 1:1 tagged
configure v103 add ports 1:3 tagged
configure v103 add ports 1:4 tagged
create vlan v104
configure v104 tag 104
configure v104 add ports 1:2 tagged
configure v104 add ports 1:3 tagged
configure v104 add ports 1:4 tagged
create vlan v101
configure v101 tag 101
configure v101 add ports 1:3 tagged
configure v101 add ports 1:4 tagged
create vlan v102
configure v102 tag 102
configure v102 add ports 1:3 tagged
configure v102 add ports 1:4 tagged
create stpd stp1
configure stp1 tag 101
configure stp1 add vlan v101
configure stp1 add vlan v102
configure stp1 add vlan v103
configure stp1 add vlan v104
enable stpd stp1
```

This set of configuration commands creates the member VLANs and enables STP on SW3:

```
create vlan v101
configure v101 tag 101
configure v101 add ports 1:3 tagged
configure v101 add ports 1:4 tagged
create vlan v102
configure v102 tag 102
configure v102 add ports 1:3 tagged
configure v102 add ports 1:4 tagged
create vlan v103
configure v103 tag 103
configure v103 add ports 1:3 tagged
configure v103 add ports 1:4 tagged
create vlan v104
configure v104 tag 104
configure v104 add ports 1:3 tagged
configure v104 add ports 1:4 tagged
create stpd stp1
configure stp1 tag 101
configure stp1 add vlan v101
configure stp1 add vlan v102
configure stp1 add vlan v103
configure stp1 add vlan v104
enable stpd stp1
```

The final set of configuration commands creates the translation VLAN and enables VLAN translation on SW3:

```
create vlan v1000
configure v1000 tag 1000
configure v1000 add ports 2:1 tagged
configure v1000 vlan-translation add member-vlan v101
configure v1000 vlan-translation add member-vlan v102
```

```
configure v1000 vlan-translation add member-vlan v103
configure v1000 vlan-translation add member-vlan v104
```

The following configuration commands create the translation VLAN and enable VLAN translation on VLANs that have overlapping ports:

```
configure v1000 vlan-translation add member-vlan v102 loopback-port 1:22
configure v1000 vlan-translation add member-vlan v103 loopback-port 1:23
configure v1000 vlan-translation add member-vlan v104 loopback-port 1:24
```

Port-Specific VLAN Tag

The Port-specific *VLAN* feature adds a layer of specificity between the port tag and the VLAN/VMAN tag: a port-specific VLAN tag. This feature adds the following functionality to the existing VLAN:

- Ability to associate a tag to a VLAN port. This tag is used as a filter to accept frames with matching VID. It is also used as the tag of the outgoing frames.
- Ability to add multiple VLAN ports on the same physical port as long as those VLAN ports are associated with different tags.
- Allows the existing untagged and tagged VLAN ports to be part of the VLAN.
- Ability to learn MAC address on port, tag and VLAN instead of only on the port.As a consequence of the previous point, ability to add static MAC address to port, tag and VLAN.
- Ability to specify limit-learning and MAC lockdown on a port, tag and VLAN, instead of only on the port.
- Rate limiting and counting of frames with matching VIDs is supported with the existing *ACL (Access Control List)*.

The Port-specific VLAN tag allows tagged VLAN ports to be configured with tag values. When the tag is not configured, it is implicit that the tag of the tagged port is the tag of the VLAN. We call the tag of the port the "port tag", and the tag of the VLAN the "base tag". The port tag is used to determine the eligibility of the frames allowed to be part of the VLAN. Once the frame is admitted to the VLAN port, the base tag is used. From a functional standpoint, the frame tag is rewritten to the base tag.

The base tag then is translated to the port tag for the outgoing frame.

Note
The port tag is equal to the base tag when the port tag is not specified, so the current VLAN behavior is preserved.

Untagged VLAN ports also have port tag, which is always the same as the base tag. Outgoing frames are untagged. The untagged VLAN port always has an implicit port tag thats's always equal to the base tag. There can be only one untagged VLAN port on a physical port. It receives untagged frames, and tagged frames, and transmits only untagged frames.

A tagged VLAN port can have a port tag configured, or not. When not configured, the port tag is equal to the base tag. There can be more than one tagged VLAN port on a

physical port. It receives tagged frames with tag equals to the port tag, and transmits tagged frames with port tag.

When the VLAN is assigned to L2VPN, the base tag is the tag that is carried by the pseudo-wire when the dot1q include is enabled. It can be viewed that VPLS PW port tag is equal to the base tag. To assign a VLAN with a port-specific tag to an L2VPN, use the existing `configure vpls vpls_name add service vlan vlan_name` command.

Since every tagged VLAN port has different VIDs, forwarding between them on the same physical port (hairpin switching) is possible. From the external traffic point of view, the frame tags are rewritten from the receive port tag to the transmit port tag. Since each port tag is a different VLAN port, a frame that has to be broadcasted to multiple VLAN ports is sent out multiple times with different tags when the VLAN ports are on the same physical port. Each port + port tag is an individual VLAN port.

MAC addresses are learned on the VLAN port. This means that the port in the *FDB* entry is the port + port tag. A unicast frame destined to a MAC address that is in the FDB is sent out of the associated VLAN port. As mentioned earlier, there is only one MAC addressed learned on the VLAN. If the MAC address is learned on a different port or a different tag, it is a MAC move. It is transmitted out of the physical port only on the associated VLAN port tagged with the port tag when the VLAN port is tagged.

When there are multiple tagged VLAN ports on the transmit port, only one frame with the right tag is transmitted. It is transmitted untagged on an untagged VLAN port. Accordingly, the static MAC address is configured on a VLAN port. This means that the port tag is specified when the tag is not equal to the base tag. The command to flush FDB does not need to change. But, a VLAN port-specific flush needs to be implemented to handle the case when a VLAN port is deleted. This flush is internal and not available through the CLI.

Per VLAN port (port + tag) rate limiting and accounting is achieved by the existing ACL. Use match condition `vlan-id` to match the port VID. You can use action `count` and `byte-count` for accounting. And you can use `show access-list counter` to view the counters. Action meter can be used for rate limiting. To create a meter, use the `create meter` command, and configure the committed rate and maximum burst size.

Supported Platforms

Port-Specific VLAN Tag is supported on the following platforms:

- ExtremeSwitching X460-G2 (supported from ExtremeXOS 15.6)
- ExtremeSwitching X690

Port-Specific Tags in L2VPN

You can assign a *VLAN* with port specific tag to VPLS/VPWS using the `configure vpls vsi add service vlan vl` command. Because this is a single VLAN, the base VID is used when dot1q include is enable. For example, when VLAN 100 that has ports on

Ethernet port 1 with port tag 10 and 11 is assigned to L2VPN, the tag that is carried by the pseudo wire is 100. The configuration for this example is as follows:

```
create vlan exchange tag 100
config vlan exchange add ports 1 tagged 10
config vlan exchange add ports 1 tagged 11
config vpls vsi1 add service vlan exchange
```

Similarly, the following is an example for VPWS. There can only be a single VLAN port in the VLAN for assignment to VPWS to be successful:

```
create vlan exchange tag 100
config vlan exchange add ports 1 tagged 10
config l2vpn vpws pw1 add service vlan exchange
```

VLAN Port State

VLAN port state is the same as the state of the Ethernet port.

ACLs

You can use the existing match vlan-id *ACL* to accomplish counting and metering. You can assign the ACL to both ingress and egress port. The followings are the examples of such configuration. The port 3 tag is 4 and the port 4 tag is 5. These ACLs will match the frame vlan-ID, and the vlan-ID specified in the match criteria is independent of the port tag.

```
Content of acl.pol
entry tag_1 {
   if {
     vlan-id 4;
     } then {
       packet-count tag_1_num_frames;
       meter tag_1_meter;
     }
}
entry tag_2 {
   if {
     vlan-id 5;
     } then {
       byte-count tag_2_num_bytes;
       meter tag_2_meter;
     }
}
Content of acl_egress.pol
entry tag_1_egr {
   if {
     vlan-id 4;
     } then {
       packet-count tag_1_egr_num_frames;
       meter tag_1_egr_meter;
     }
}
entry tag_2_egr {
   if {
     vlan-id 5;
     } then {
       byte-count tag_2_egr_num_bytes;
       meter tag_2_egr_meter;
     }
}
```

Configuring Port-Specific VLAN Tags

The following specific commands are modified by the port-specific *VLAN* tag:

- `clear fdb`: Only clears on physical port or VLAN, not on a vlan port.
- `delete fdb`: All or specific MAC address, or specific MAC address on a VLAN.
- `enable/disable flooding ports`: Only on physical port (applies to all VLAN ports).
- `enable/disable learning`: Only on physical port (applies to all VLAN ports on the same physical port), or on a VLAN (applies to all VLAN ports of the VLAN).
- `show fdb stats`: Only on physical port or VLAN, not on a VLAN port.

Use the following commands to configure Port-specific VLAN tags:

- To configure the port-specific tag, use the `configure ports` *`port_list`* {**`tagged`** *`tag`*} **`vlan`** *`vlan_name`* [**`limit-learning`** *`number`* {**`action`** [**`blackhole`** | **`stop-learning`**]} |**`lock-learning`** | **`unlimited-learning`** | **`unlocklearning`**] command.
- To specify the port tag when you need to put multiple vlans into a broadcast domain, use the `configure` {**`vlan`**} *`vlan_name`* `addports` [**`port_list`** | **`all`**] {**`tagged`**{*`tag`*} | **`untagged`**} {{**`stpd`**} *`stpd_name`*} {**`dot1d`** | **`emistp`** | **`pvst-plus`**}} command.
- To specify a port tag to delete a VLAN port that has a different tag from the VLAN tag, use the `configure` {**`vlan`**} *`vlan_name`* `deleteports` [**`all`** | **`port_list`** {**`tagged`** *`tag`*}] command.
- To display output of a vlan that has a port-specific tag, use the `show vlan` command.
- To display port info that has port-specific tag statistics, use the `show port` info detail command.
- To adds a permanent, static entry to the *FDB*, use the `create fdb` **`mac_addr vlan`** *`vlan_name`* [**`ports`** *`port_list`* {**`tagged`** *`tag`*} | **`blackhole`**] command.
- To show output where the port tag is displayed, use the `show fdb` command.

VMAN (PBN)

VMAN Overview on page 2261
VMAN Configuration Options and Features on page 2267
Configuration on page 2270
Displaying VMAN Information on page 2273
Configuration Examples on page 2273

The virtual metropolitan area network (VMAN) feature allows you to scale a Layer 2 network and avoid some of the management and bandwidth overhead required by Layer 3 networks.

Note
If a failover occurs, VMAN operation is not interrupted. The system has hitless failover—network traffic is not interrupted during a failover.

VMAN Overview

The VMAN feature is defined by the IEEE 802.1ad standard, which is an amendment to the IEEE 802.1Q *VLAN (Virtual LAN)* standard.

A VMAN is a virtual Metropolitan Area Network (MAN) that operates over a physical MAN or Provider Bridged Network (PBN). This feature allows a service provider to create VMAN instances within a MAN or PBN to support individual customers. Each VMAN supports tagged and untagged VLAN traffic for a customer, and this traffic is kept private from other customers that use VMANs on the same PBN.

The PBN uses Provider Bridges (PBs) to create a Layer 2 network that supports VMAN traffic. The VMAN technology is sometimes referred to as VLAN stacking or Q-in-Q.

Note
Virtual MAN (VMAN) is an Extreme Networks term that became familiar to Extreme Networks customers before the 802.1ad standard was complete. The term VMAN is used in the ExtremeXOS software and also in this document to support customers who are familiar with the term. The term PBN is also used in this guide to establish the relationship between this industry standard technology and the Extreme Networks VMAN feature.

Figure 253 shows a VMAN, which spans the switches in a PBN.

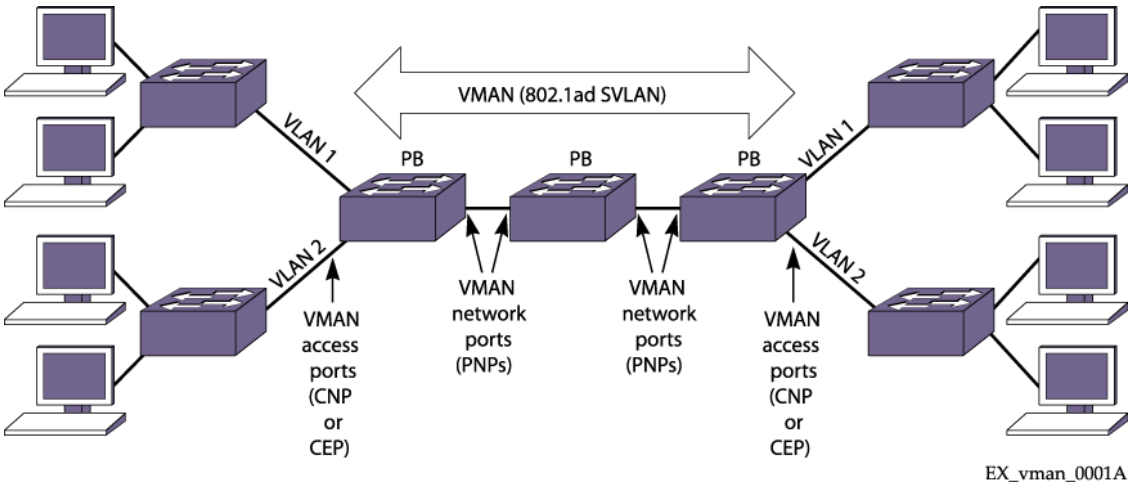


Figure 253: VMAN in PBN

The entry points to the VMAN are the access ports on the VMAN edge switches. Customer VLAN (CVLAN) traffic that is addressed to locations at other VMAN access ports enters the ingress access port, is switched through the VMAN, and exits the egress access port. If you do not configure any frame manipulation options, the CVLAN frames that exit the VMAN are identical to the frames that entered the VMAN.

VMAN access ports operate in the following roles:

- Customer Network Port (CNP)
- Customer Edge Port (CEP, which is also known as Selective Q-in-Q)

The CEP role, which is configured in software as a cep vman port, connects a VMAN to specific CVLANs based on the CVLAN CVID. The CNP role, which is configured as an untagged vman port, connects a VMAN to all other port traffic that is not already mapped to the port CEP role. These roles are described later.

All other VMAN ports (except the access ports) operate as VMAN network ports, which are also known as Provider Network Ports (PNPs) in the 802.1ad standard. The VMAN network ports connect the PBs that form the core of the VMAN. During configuration, the VMAN network ports are configured as tagged VMAN ports.

Figure 254 on page 2263 shows one VMAN, but a PBN can support multiple VMAN instances, which are sometimes called VMANs or Service VLANs (SVLANs). VMANs allow you to partition the PBN for customers in the same way that VLANs allow you to partition a Layer 2 network. For example, you can use different VMANs to support different customers on the PBN, and the PBN delivers customer traffic only to the PBN ports that are configured for appropriate VMAN.

A VMAN supports two tags in each Ethernet frame, instead of the single tag supported by a VLAN Ethernet frame. The inner tag is referred to as the customer tag (C-tag), and this optional tag is based on the CVLAN tag if the source VLAN is a tagged VLAN. The outer tag is referred to as the service tag (S-tag) or VMAN tag or SVLAN tag, and it is the tag that defines to which SVLAN a frame belongs. Figure 254 on page 2263 shows the frame manipulation that occurs at the VMAN edge switch.

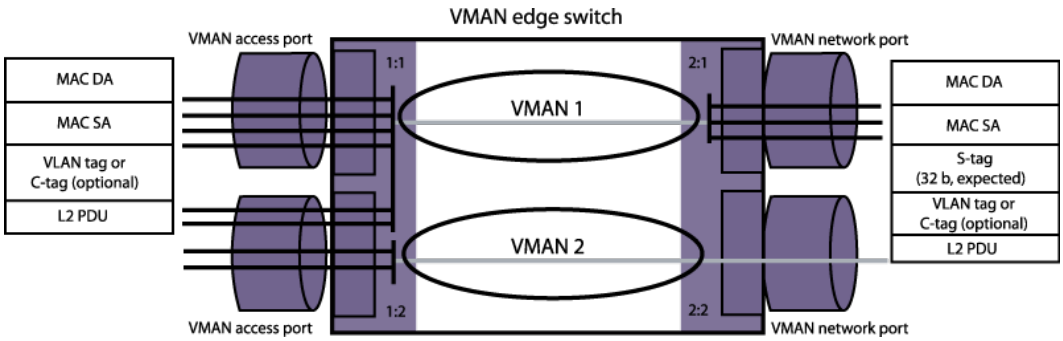


Figure 254: Tag Usage at the VMAN Access Switch

In this example, the switch accepts CVLAN frames on VMAN access ports 1:1 and 1:2. The switch then adds the S-tag to the frames and switches the frames to network ports 2:1 and 2:2. When the 802.1ad frames reach the PB egress port, the egress switch removes the S-tag, and the CVLAN traffic exits the egress access port in its original form.

When the switch in the figure above acts as the egress switch for a VMAN, VMAN frames arrive on network ports 2:1 and 2:2. The switch accepts only those frames with the correct S-tag, removes the S-tags, and switches those frames to access ports 1:1 and 1:2. Unless special configuration options are applied, the egress frames are identical to ingress CVLAN frames. (Configuration options are described in VMAN Configuration Options and Features on page 2267.)

Figure 255 shows that the S-tags and C-tags used in VMAN frames contain more than just customer and service VLAN IDs.

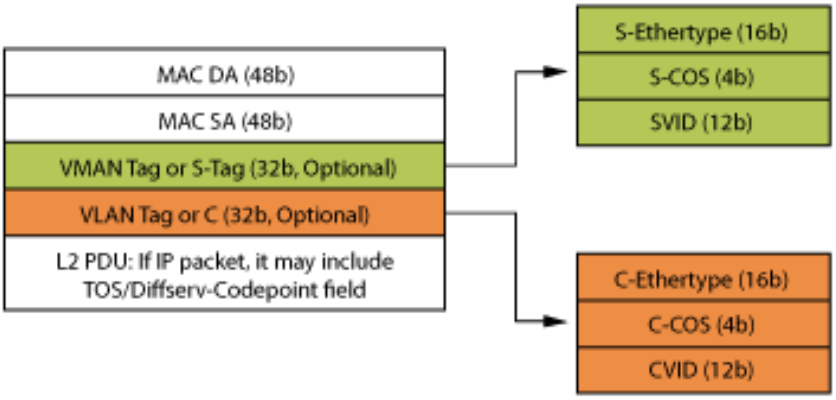


Figure 255: S-tag and C-tag Components

Each S-tag and C-tag contains an ethertype, a Class of Service (CoS), and a SVLAN ID (SVID) or CVLAN ID (CVID). The ethertype is described in Secondary Ethertype Support on page 2268, and the CoS is described in QoS Support on page 2269.

The SVID is the VLAN tag you assign to a VMAN when you create it (see the `configure vman vman_name tag tag` command. The CVID represents the CVLAN tag for tagged VLAN traffic.

Switch ports support VMAN roles and features, which are described in the following sections:

- Customer Network Ports
- Customer Edge Ports
- CVID Translation
- CVID Egress Filtering

Customer Network Ports

Customer Network Ports (CNPs) are edge switch ports that accept all tagged and untagged CVLAN traffic and route it over a single VMAN. A CNP is simpler to configure than a CEP, because it supports one VMAN on a physical port and requires no configuration of CVIDs. The VMAN service provider does not need to know anything about the CVLAN traffic in the VMAN. The service provider simply manages the VMAN, and the ingress CVLAN traffic is managed by the customer or another service provider. This separation of CVLAN and VMAN management reduces the dependence of the separate management teams on each other, and allows both management teams to make changes independent of the other.

Note
The CNP term is defined in the IEEE 802.1ad standard and is also called a port-based service interface. The CNP operation is similar to a MEF 13 UNI Type 1.2, and in releases before ExtremeXOS 12.6, CNPs were known as VMAN access ports or untagged vman ports. With the addition of CEPs, the term VMAN access port is now a generic term that refers to CNPs and CEPs.

A PBN can support up to 4094 VMANs, and each VMAN can support up to 4094 CVLANs. Because each CNP connects to only one VMAN, the maximum number of customer VMANs on an edge switch is equal to the total number of switch ports minus one, because at least one port is required to serve as the PNP (Provider Network Port).

Customer Edge Ports

Each *CEP (Customer Edge Port)* supports the configuration of connections or mappings between individual CVLANs and multiple VMANs.

This provides the following benefits:

- Each physical port supports multiple customers (each connecting to a separate VMAN).
- Each switch supports many more customer VMANs using CEPs instead of CNPs.

To define the connections between CVLANs and SVLANs, each CEP uses a dedicated CVID map, which defines the supported CVIDs on the CEP and the destination VMAN for each CVID. For example, you can configure a CEP to forward traffic from five specific

CVLANs to VMAN A and from ten other specific CVLANs to VMAN B. During VMAN configuration, certain ports are added to the VMAN as CEPs, and certain CVIDs on those ports are mapped to the VMAN. To enable customer use of a VMAN, service providers must communicate the enabled CVIDs to their customers. The customers must use those CVIDs to access the VMAN.

Note

The *CEP* term is defined in the IEEE 802.1ad standard and is also called a *C-tagged service interface*. The CEP operation is similar to a MEF 13 UNI Type 1.1.

Assigning Port CVID on Port-based or CEP VMAN Service

As of ExtremeXOS 21.1, an optional port CVID parameter has been added to the existing untagged and CEP VMAN port configuration options. When present, any untagged packet received on the port is double tagged with the configured port CVID and the SVID associated with the VMAN. If the port is untagged, packets received with a single CID still have the SVID added. If the port is CEP, only untagged and any specifically configured CVIDs are allowed. As double tagged ports are received from tagged VMAN ports and forwarded to untagged VMAN ports, the SVID associated with the VMAN is stripped. Additionally, the CVID associated with the configured port CVID is also stripped in the same operation. If the port is CEP and CEP egress filtering in enabled, only the specified port CVID and CVIDs are allowed to egress.

Much like the CVIDs configured as part of the CEP feature, the configured port CVID is not represented by a *VLAN* within ExtremeXOS. The implication is that protocols and individual services cannot be applied to the Port CVID alone. Protocols and services are instead applied to the VMAN and/or port as the VMAN represents the true layer-2 broadcast domain. Much like regular untagged and CEP VMAN ports, MAC *FDB (forwarding database)* learned occurs on the VMAN so duplicate MAC addresses received on multiple CVIDs which are mapped to the same VMAN can be problematic.

Even when the additional Port CVID is configured, the port still has all of the attributes of a regular untagged VMAN port. This means that any single c-tagged packets received on the same port will have just the SVID associated with the VMAN added to the packet. Likewise, any egress packet with a CVID other than the configured Port CVID will have just the SVID stripped. This is not true when the Port CVID specification is part of the CEP configuration, added in ExtremeXOS 16.2.1.

Coexistence with Tagged VLANs Interfaces, CEP VMAN Interfaces, and Tagged VMAN Interfaces

Since the port-cvid configuration still has the attributes of a regular untagged VMAN, all of the *VLAN* and VMAN exclusion and compatibility rules of a regular untagged VMAN port also apply. A list of these rules is contained in “EXOS Selective Q-in-Q”. In ExtremeXOS 21.1, when port-cvid is specified as part of CEP, there can only be one such port-cvid on the port. Any adjacent CEP CVIDs will not be included in the same CVID range.

Protocol and Feature Interactions

This feature leverages existing untagged and CEP VMAN port infrastructure, therefore any protocol that works with a regular untagged and CEP VMAN port will also work when the optional Port CVID is additionally configured.

Protocols that locally originate control packets, such as *STP (Spanning Tree Protocol)* and ELRP which are used for loop prevention, will transmit packets as natively untagged on the wire when the port is an untagged VMAN member. These untagged packets will also be able to be received and processed by ExtremeXOS. This makes STP edge safeguard + BPDU guard or ELRP effective ways to detect and react to network loops on the device. However, since control packets are transmitted as untagged upstream devices may need additional configuration support to properly detect remote loops not directly attached to the device.

Other effective loop prevention mechanisms work without any interaction with untagged VMAN ports. For example, turning physical port auto-polarity off will prevent an accidental looped cable from becoming active. Likewise, storm-control rate limiting of broadcast and flood traffic can be applied in this environment to minimize the effects of a network loop. In addition to detecting, preventing, and minimizing the effects of a network loop, user *ACL (Access Control List)*s can be applied to gain visibility and control of L2, L3, and L4 match criteria even with double tagged packets. All applicable ACL action modifiers are available in this environment. IP multicast pruning within a VMAN can be accomplished via normal *IGMP (Internet Group Management Protocol)* snooping. ExtremeXOS supports full IGMP snooping and IP multicast pruning of single tagged and double tagged packets. However, when an IP address is configured on the VMAN, the IGMP protocol engine will transmit single tagged packets on tagged VMAN ports or untagged packets on untagged VMAN ports so upstream switch configuration and support may be necessary to properly propagate group memberships across the network.

CVID Translation

To support CVLANs that are identified by different CVIDs on different CEPs, some switches support a feature called CVID translation, which translates the CVID received at the VMAN ingress to a different CVID for egress from the VMAN (see Figure 256).

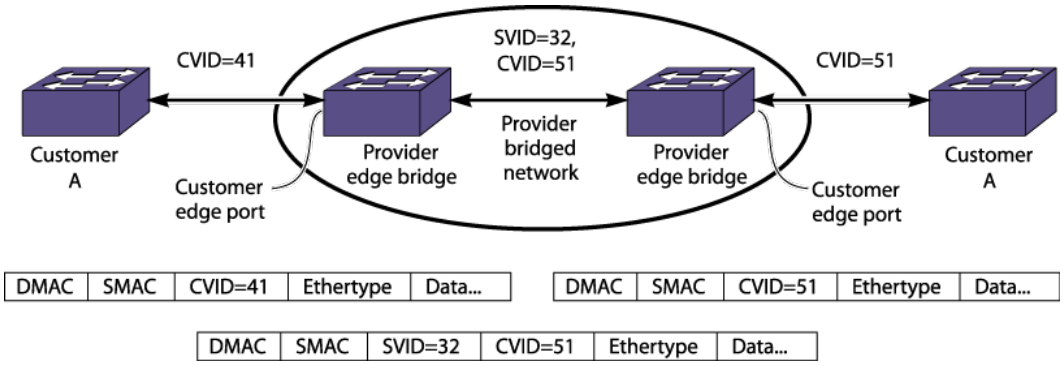


Figure 256: CVID Translation

You can use a CLI command to configure CVID translation for a single CVID or for a range of CVIDs, and you can enter multiple commands to define multiple CVIDs and

ranges for translation. The commands can be applied to a single port or a list of ports, and after configuration, the configuration applied to a port is retained by that port.

Note
CVID translation is available only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

CVID translation can reduce the number of CVIDs that can be mapped to VMANs.

CVID Egress Filtering

CVID egress filtering permits the egress from VMAN to CEP of only those frames that contain a CVID that has been mapped to the source VMAN; all other frames are blocked. For example, Customer Edge Port A in Figure 257 is configured to support CVIDs 10-29, and Customer Edge Port B is configured to support CVIDs 10-19. If CVID egress filtering is enabled on Customer Edge Port B, frames with CVIDs 20-29 will not be forwarded at the egress of Customer Edge Port B.

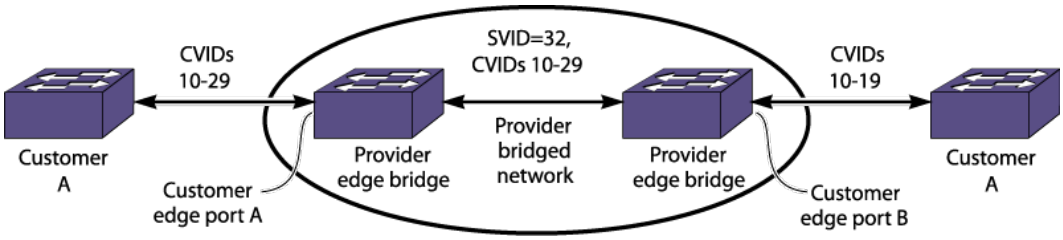


Figure 257: CVID Egress Filtering

You can enable CVID egress filtering for a single CEP or for all CEPs with a CLI command. You can also repeat the command to enable this feature on multiple CEPs.

Note
CVID egress filtering is available only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.

When this feature is enabled, it reduces the maximum number of CVIDs that can be mapped to VMANs. The control of CVID egress filtering applies to fast-path forwarding. When frames are forwarded through software, CVID egress filtering is always enabled.

VMAN Configuration Options and Features

As of ExtremeXOS 16.1, you can now referr to a VMAN by VID as an alternative. Specifying lists of VIDs is a is a useful shortcut to all users that can reduce the number of commands required to configure the switch.

Note
Commands enhanced with VID list support operate in a “best effort” fashion. If one of the VIDs in a VID list do not exist the command is still executed for all of the VIDs in the list that do exist. No error or warning is displayed for the invalid VIDs unless all of the specified VIDs are in valid.

ACL Support

The ExtremeXOS software includes VMAN (PBN) *ACL* support for controlling VMAN frames.

VMAN ACLs define a set of match conditions and modifiers that can be applied to VMAN frames. These conditions allow specific traffic flows to be identified, and the modifiers allow a translation to be performed on the frames.

Secondary Ethertype Support

The C-tag and S-tag components that are added to all VMAN (PBN) frames include C-ethertype and S-ethertype components that specify an ethertype value for the customer *VLAN* and VMAN, respectively.

Note
This feature is supported only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

The C-tag and S-tag components that are added to all VMAN (PBN) frames include C-ethertype and S-ethertype components that specify an ethertype value for the customer VLAN and VMAN, respectively. The I-tag used in PBBN frames also includes an ethertype value. When a VLAN or VMAN frame passes between two switches, the ethertype is checked for a match. If the ethertype does not match that of the receiving switch, the frame is discarded.

The default ethertype values are:

- VLAN port (802.1q frames): 0x8100
- Primary VMAN port (802.1ad frames): 0x88A8
- Secondary VMAN port (802.1ad frames): Not configured

The secondary ethertype support feature applies only to VMANs. The ethertype value for VLAN frames is standard and cannot be changed.

If your VMAN transits a third-party device (in other words, a device other than an Extreme Networks device), you must configure the ethertype value on the Extreme Networks device port to match the ethertype value on the third-party device to which it connects.

The secondary ethertype support feature allows you to define two ethertype values for VMAN frames and select either of the two values for each port.

For example, you can configure ports that connect to other Extreme Networks devices to use the default primary ethertype value, and you can configure ports that connect to other equipment to use the secondary ethertype value, which you can configure to match the requirements of that equipment.

When you create a VMAN, each VMAN port is automatically assigned the primary ethertype value. After you define a secondary ethertype value, you can configure a port to use the secondary ethertype value. If two switch ports in the same VMAN use different ethertype values, the switch substitutes the correct value at each port. For

example, for VMAN edge switches and transit switches, the switch translates an ingress ethertype value to the network port ethertype value before forwarding. For egress traffic at VMAN edge switches, no translation is required because the switch removes the S-tag before switching packets to the egress port.

You can set the primary and secondary ethertypes to any value, provided that the two values are different.

QoS Support

The VMAN (PBN) feature interoperates with many of the *QoS (Quality of Service)* and HQoS features supported in the ExtremeXOS software.

One of those features is egress queue selection, which is described in the next section. For more information on other QoS and HQoS features that work with VMANs, see Quality of Service on page 1453.

Egress Queue Selection

This feature examines the 802.1p value or Diffserv code point in a VMAN (PBN) S-tag and uses that value to direct the packet to the appropriate queue on the egress port.

Note
This feature is supported only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

On some systems you can configure this feature to examine the values in the C-tag or the S-tag. For more information, see the list in the Switch Engine v33.6.1 Licensing Guide document.. For instructions on configuring this feature, see Selecting the Tag used for Egress Queue Selection on page 2273.

VMAN Double Tag Support

The VMAN double tag feature adds an optional port CVID parameter to the existing untagged VMAN port configuration. When present, any untagged packet received on the port will be double tagged with the configured port CVID and SVID associated with the VMAN. Packets received with a single CVID on the same port will still have the SVID added. As double tagged packets are received from tagged VMAN ports and forwarded to untagged VMAN ports, the SVID associated with the VMAN is stripped. Additionally, the CVID associated with the configured Port CVID is also stripped in the same operation.

Much like the CVIDs configured as part of the CEP feature, the configured Port CVID is not represented by a *VLAN* within ExtremeXOS. The implication is that protocols and individual services cannot be applied to the Port CVID alone. Protocols and services are instead applied to the VMAN and/or port as the VMAN represents the true layer-2 broadcast domain. Much like regular untagged VMAN ports, MAC *FDB* learning occurs on the VMAN, so duplicate MAC addresses received on multiple CVIDs that are mapped to the same VMAN can be problematic. Even when the additional Port CVID is configured, the port still has all of the attributes of a regular untagged VMAN port. This

means that any single c-tagged packets received on the same port will have just the SVID associated with the VMAN added to the packet. Likewise, any egress packet with a CVID other than the configured Port CVID will have the SVID stripped.

See Coexistence with Tagged VLANs Interfaces, CEP VMAN Interfaces, and Tagged VMAN Interfaces on page 2265.

See Protocol and Feature Interactions on page 2265.

Configuration

Configuring VMANs (PBNs)

Guidelines for Configuring VMANs

The following are VMAN configuration guidelines for all platforms:

- Duplicate customer MAC addresses that ingress from multiple VMAN access ports on the same VMAN can disrupt the port learning association process in the switch.
- VMAN names must conform to the guidelines described in Object Names on page 34.
- You must use mutually exclusive names for:
  - *VLAN*s
  - VMANs
  - IPv6 tunnels
- VMAN ports can belong to load-sharing groups.
- You can enable or disable jumbo frames before configuring VMANs. You can enable or disable jumbo frames on individual ports. See Configuring Ports on a Switch on page 223 for more information on configuring jumbo frames.
- *STP* operation on CVLAN components in a PEB as described in IEEE 802.1ad is not supported.
- The initial version of this feature does not implement an XML API.
- Multiple VMAN roles can be combined on one port with certain VLAN types as shown in Table 176.

Table 176: Port Support for Combined VMAN Roles and VLANs

Platform	Combined CNP, CEP, and Tagged VLAN [a,b]	Combined PNP, CNP, and CEP [a,b,c]	Combined PNP and Tagged VLAN	Combined PNP and Untagged VLAN
All ExtremeSwitching Universal platforms	X	X	X [d]	X

Note
If you already configured VLANs and VMANs on the same stand-alone switch using ExtremeXOS 11.4 or later, you cannot change the VMAN ethertype from 0X8100 without first removing either the VLAN or VMAN configuration.

Procedure for Configuring VMANs

This section describes the procedure for configuring VMANs. Before configuring VMANs, review Guidelines for Configuring VMANs on page 2270. To configure a VMAN, complete the following procedure at each switch that needs to support the VMAN:

1. If you are configuring a SummitStack, or an ExtremeSwitching series switch, enable jumbo frames on the switch.

   Note
   You must enable jumbo frames before configuring VMANs on these systems.

2. Create a VMAN using the command:

   ```
   create vman vman_name | vman_list vr vr_name
   ```

3. Assign a tag value to the VMAN using the command:.

   ```
   configure vman vman_name | vman_list tag tag_id
   ```

4. To configure PNP ports on a PEB or PB, use the following command with the tagged option:

   ```
   configure vman vman_name | vman_id add ports [ all | port_list ]
   {untagged | tagged}
   ```

5. To configure CNP ports on a PEB, use the following command with the untagged option:

   ```
   configure vman vman_name add ports [ all | port_list ] {untagged |
   tagged}
   ```

   Note
   You must configure CNP ports as untagged, so that the S-tag is stripped from the frame on egress. If the port-cvid is configured, any untagged packet received on the port will be double tagged with the configured port CVID and the SVID associated with the VMAN. Packets received with a single CVID on the same port will still have the SVID added as usual.
   As double tagged packets are received from tagged VMAN ports and forwarded to untagged VMAN ports,the SVID associated with the VMAN is stripped. Additionally, the CVID associated with the configured port CVID is also stripped in the same operation.

[a] Subsets of this group are also supported. That is, any two of these items are supported.
[b] When a CNP is combined with a CEP or tagged VLAN, any CVIDs not explicitly configured for a CEP or tagged VLAN are associated with the CNP.
[c] A PNP (tagged VMAN) and a CNP (untagged VMAN) or CEP cannot be combined on a port for which the selected VMAN ethertype is 0x8100.
[d] If the secondary VMAN ethertype is selected for the port, it must be set to 0x8100.

6. To configure CEP ports on a PEB, do the following:
   a. Use the following command to establish a physical port as a CEP and configure CVID mapping and translation:

      ```
      configure vman vman_name | vman_id add ports port_list cep [ cvid <cvid or cvid_range> { translate <cvid or cvid_range> } } | port-cvid <port_cvid> ] }
      ```

   b. Use the following commands to add or delete CVIDs for a CEP and manage CVID mapping and translation:

      ```
      configure vman vman_name | vman_id ports port_list add cvid {cvid | cvid_range} {translate cvid | cvid_range }
      configure vman vman_name ports port_list delete cvid {cvid | cvid_range }
      ```

   c. Use the following commands to manage CVID egress filtering for a CEP:

      ```
      enable vman cep egress filtering ports {port_list | all}
      disable vman cep egress filtering ports {port_list | all}
      ```

7. Configure additional VMAN options as described in Configuring VMAN Options on page 2272.
8. To configure a *VLAN* to use a VMAN, configure the VLAN on the switch port at the other end of the line leading to the VMAN access port.

Configuring VMAN Options

Configuring the Ethertype for VMAN Ports

The ethertype is a component of *VLAN* and VMAN frames. It is introduced in Secondary Ethertype Support on page 2268.

Note
This feature is supported only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

To configure the ethertype for VMAN (PBN) ports, do the following:

1. Configure the primary and secondary (if needed) VMAN ethertype values for the switch using the following command:

   ```
   configure vman ethertype hex_value [primary | secondary]
   ```

   By default, all VMAN ports use the primary ethertype value.
2. If you plan to use a secondary ethertype, select the secondary ethertype for the appropriate VMAN ports using the following command:

   ```
   configure port port_list ethertype {primary | secondary}
   ```

Selecting the Tag used for Egress Queue Selection

By default, switches that support the enabling and disabling of this feature use the 802.1p value in the S-tag to direct the packet to the queue on the egress port.

Note
This feature is supported and configurable only on the platforms listed for this feature in the license tables in the Switch Engine v33.6.1 Licensing Guide document.

- Configure egress queue dot1p examination of the C-tag using:
  `enable dot1p examination inner-tag port [all | port_list]`
- Return to the default selection of using the 802.1p value in the S-tag using:
  `disable dot1p examination inner-tag ports [all | port_list]`

Note
See Quality of Service on page 1453 for information on configuring and displaying the current 802.1p and DiffServ configuration for the S-tag 802.1p value. To enable dot1p examination for inner-tag, dot1p examination for outer-tag must be disabled using the command `disable dot1p examination ports [all | port_list]`

Displaying VMAN Information

Use the following commands to display information on one or all VMANs.

show {**vman**} *vman_name* {**ipv4** | **ipv6**}

show vman [**tag** *tag_id* | **detail**] {**ipv4** | **ipv6**}

show {**vman**} *vman_name* **eaps**

Note
The display for the `show vman` command is different depending on the platform and configuration you are using. See the Switch Engine v33.6.1 Command References for complete information on this command.

You can also display VMAN information, as well as all the VLANs, by issuing the `show ports information detail` command. And you can display the VMAN ethernet type and secondary etherType port_list by using the `show vman` **etherType** command

Configuration Examples

VMAN CEP Example

The following configuration configures a VMAN CEP to support up to 10 customer VLANs for each of three VMANs.

```
create vman cust1
```

```
create vman cust2
create vman cust3
config vman cust1 tag 1000
config vman cust2 tag 1001
config vman cust3 tag 1002
config vman cust1 add port 22 tag
config vman cust2 add port 22 tag
config vman cust3 add port 23 tag
config vman cust1 add port 1 cep cvid 100 - 109
config vman cust2 add port 1 cep cvid 110 - 119
config vman cust3 add port 1 cep cvid 120 - 129
enable vman cep egress filtering ports 1
```

Port 1 serves as the CEP, and egress filtering is enabled on the port. Ports 22 and 23 serve as CNPs, providing the connection between the CEP port and the rest of each VMAN.

Multiple VMAN Ethertype Example

Figure 258 shows a switch that is configured to support the primary ethertype on three ports and the secondary ethertype on a fourth port.

The primary VMAN (PBN) ethertype is changed from the default value, but that is not required.

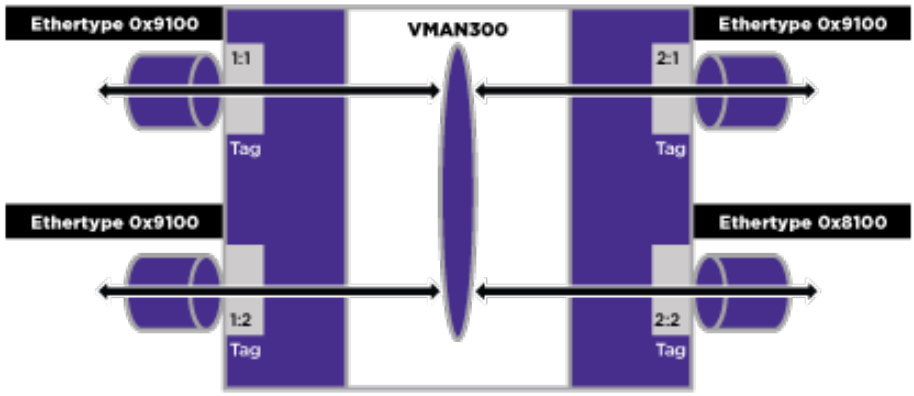
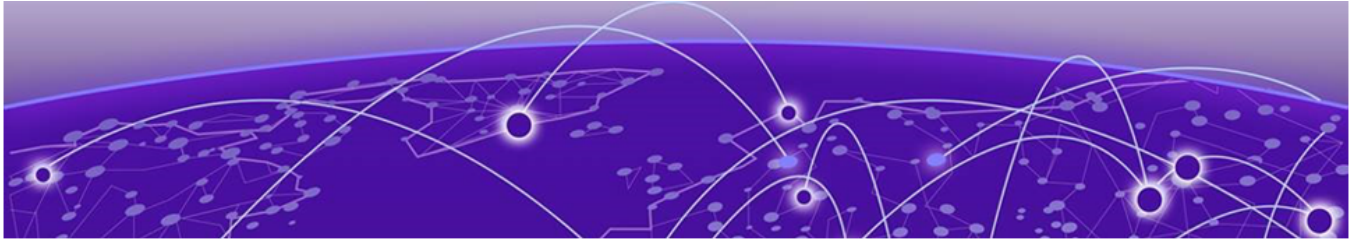


Figure 258: Multiple VMAN Ethertype Example

The following configuration commands accomplish what is shown above:

```
# configure vman ethertype 0x9100 primary
# configure vman ethertype 0x8100 secondary
#
# configure port 2:2 ethertype secondary
#
# create vman vman300
# configure vman vman300 tag 300
```

```
#
# configure vman vman300 add port 1:1, 2:1, 2:2 tagged
# configure vman vman300 add port 1:2 untagged
```



VRRP

VRRP Overview on page 2276
Configuring VRRP on page 2297
Managing VRRP on page 2301
Displaying VRRP Information on page 2301
VRRP Configuration Examples on page 2302

This chapter assumes that you are already familiar with the Virtual Router Redundancy Protocol (VRRP). If not, refer to the following publications for additional information:

- RFC 2338—Virtual Router Redundancy Protocol (VRRP)
- RFC 2787—Definitions of Managed Objects for the Virtual Router Redundancy Protocol
- RFC 3768—Virtual Router Redundancy Protocol (VRRP) Version 2 for IPv4 www.ietf.org/rfc/rfc3768.txt
- RFC 5798—Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 http://tools.ietf.org/html/rfc5798
- Draft IETF VRRP Specification v2.06 http://tools.ietf.org/html/draft-ietf-vrrp-spec-v2-06

VRRP Overview

VRRP (Virtual Router Redundancy Protocol), like the *ESRP (Extreme Standby Router Protocol)*, allows multiple switches to provide redundant routing services to users.

VRRP is used to eliminate the single point of failure associated with manually configuring a default gateway address on each host in a network. Without using VRRP, if the configured default gateway fails, you must reconfigure each host on the network to use a different router as the default gateway. VRRP provides a redundant path for the hosts. Using VRRP, if the default gateway fails, the backup router assumes forwarding responsibilities. An example VRRP topology is shown in Figure 259.

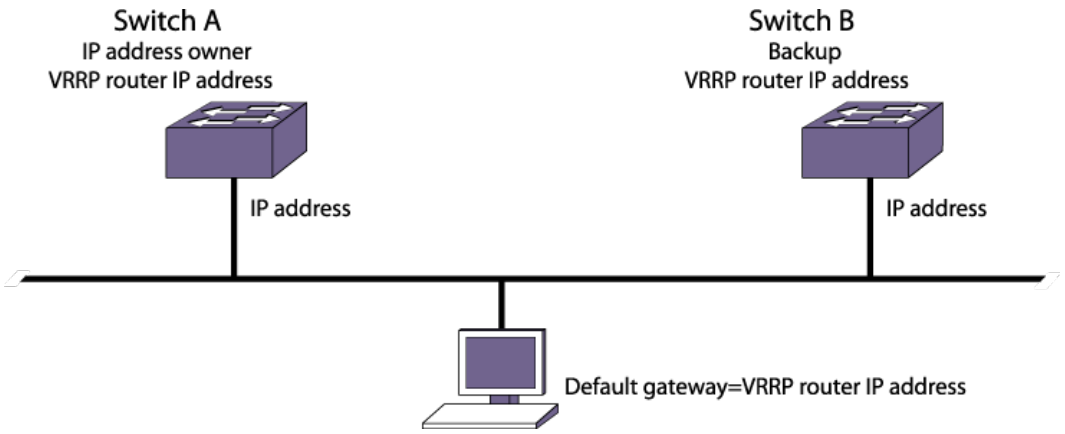


Figure 259: Simple VRRP Network

Switches A and B are both configured with the same VRRP router ID on the same *VLAN (Virtual LAN)*, which establishes a VRRP relationship between the two routers. Because a single switch can support multiple VRRP relationships, each relationship is referred to as a *VRRP router instance*. Within a VRRP router instance, any VRRP router can become the master, but only one VRRP router can be master at a time. The master processes all client communications, and the other VRRP routers in the VRRP routing instance stand by, ready to take over if the master is no longer available.

Each switch in a VRRP topology has its own unique IP and MAC addresses, which are required for basic IP connectivity. For each VRRP router instance, there are shared VRRP IP and MAC addresses, which are used for network client communications. The VRRP router IP address is configured on all VRRP routers in a VRRP routing instance, and it is configured as the default gateway address on network clients. If the master VRRP router becomes unavailable, the backup VRRP router takes over using the same VRRP router IP address.

If the VRRP router IP address matches the actual VLAN IP address of the IP address owner has the highest priority value (255) and will always become the master when VRRP is enabled and operating correctly. If the switch or the VRRP process on the switch stops responding, a backup switch (Switch B in Figure 260) takes over the master role and serves as the default gateway for network clients.

VRRP supports multiple backup routers. If the master VRRP router stops working, one of the backup routers takes over as described in ESRP Master Election on page 620.

VRRP also supports multiple VRRP router instances, which can be used to enable load sharing. The following figure shows a VRRP load-sharing configuration.

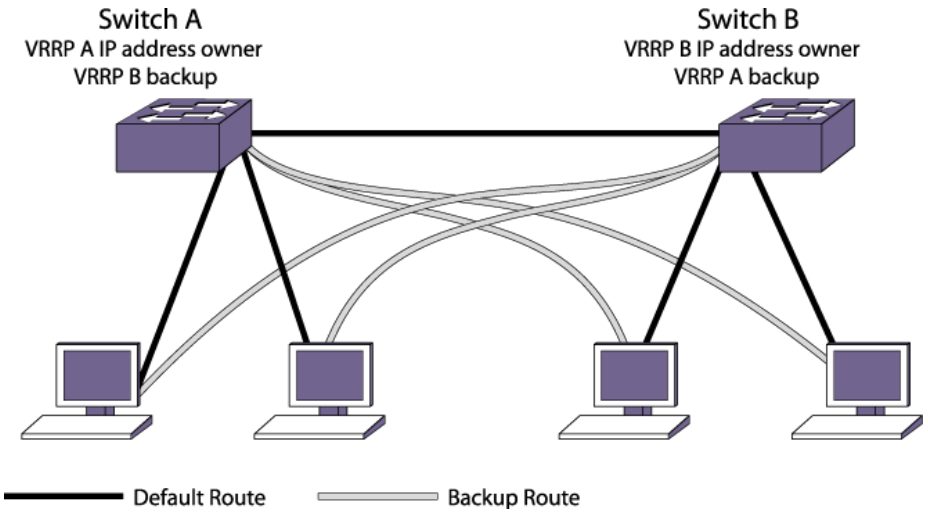


Figure 260: VRRP Load-sharing Configuration

Switches A and B in the above figure are each configured with two VRRP router instances. Switch A is the IP address owner and default master for VRRP instance 1, and Switch B is the IP address owner and default master for VRRP instance 2. Half the network clients are configured to use VRRP instance 1 as the primary gateway and VRRP instance 2 as the backup gateway. The other half of the network clients are configured to use VRRP instance 2 as the primary gateway and VRRP instance 1 as the backup gateway. When both switches are operating with VRRP, each switch supports half the clients in a load-sharing topology. If either switch fails, or if VRRP is disabled on a switch, the remaining switch supports all network clients.

Note
We recommend that you do not enable VRRP on aggregated VLANs, which are also known as super VLANs.

VRRP Guidelines

The following guidelines apply to using *VRRP*.

- VRRP packets are encapsulated IP packets.
- The VRRP IPv4 multicast address is 224.0.0.18.
- The VRRP IPv6 multicast address is ff02::12.
- The maximum number of supported VRIDs per interface is 255 for V3-v2 version mode, and 255 for v2 or v# version mode.
- An interconnect link between VRRP routers should not be used, except when VRRP routers have hosts directly attached.
- VRRP instance scale is dependent on platform and license. Please refer to the ExtremeXOS release notes for details.
- Up to 255 unique VRIDs can be configured on the router.
- VRRP and other L2 redundancy protocols can be simultaneously enabled on the same switch.

- When VRRP and BOOTP/*DHCP (Dynamic Host Configuration Protocol)* relay are both enabled on the switch, the relayed BOOTP agent IP address is the actual switch IP address, not the virtual IP address.
- We do not recommend simultaneously enabling VRRP and *ESRP* on the same switch.
- VRRP and ESRP cannot be configured on the same *VLAN* or port. This configuration is not allowed.
- RFC 5798 describes a situation where a master VRRP router takes on a duplicate IP address due to interaction with the duplicate address detection (DAD) feature. To prevent such duplicate addresses, the DAD feature is disabled whenever a VRRP router is configured for IPv6 or IPv4.
- A VRRP router instance can be configured with multiple IP addresses on the same subnet or on different subnets, provided that all virtual IP addresses match the subnet address of a VLAN on the switch. For example, if a host switch has VLAN IP addresses in the 1.1.1.x and 2.2.2.x subnets, then that VRRP router instance can contain virtual IP addresses in both those subnets as well.
- If a VRRP router instance is assigned priority 255, then the host router must own all the IP addresses assigned to the VRRP router instance. That is, each virtual IP address must match an IP address configured for a VLAN on the router.
- When a VRRPv2 instance spans routers using ExtremeXOS version 12.6 and earlier and routers using ExtremeXOS version 12.7 and later, routers using ExtremeXOS version 12.6 and earlier log packet-size warning messages.
- VRRP scaling numbers differs based on the license and hardware used; please refer the release notes for individual scaling limits.

Note
A maximum of 511 VRRP instances are allowed without One-To-Many Mirroring. If using OTM Mirroring, only 507 VRRP instances are allowed. You can increase VRRP scaling limit by using VRRP grouping (see VRRP Grouping to Increase VR Scaling on page 2292).

Note
The behavior of ExtremeXOS 30.1 has changed if locally configured IP addresses are used to determine network reachability. For details, see the `ping` command.

VRRP Master Election

When a *VRRP* configured network starts, VRRP uses an election algorithm to dynamically assign master responsibility to one of the VRRP routers on the network.

The VRRP master is determined by the following factors:

- VRRP priority—Possible values are 0 through 255. Value 255 is reserved for the VRRP router IP address owner, and value 0 is reserved for the master router, to indicate it is releasing master responsibility. Values 1–254 can be configured on backup routers to influence which backup router becomes the master when the master VRRP router

is no longer available. The higher number has higher priority. The default value for backup routers is 100.

- Higher IP address—If multiple backup routers have the same configured priority, the router with the highest IP address becomes the master.

If the master router becomes unavailable, the election process begins and the backup router that wins the election assumes the role of master.

Note
In VRRP IPv6, master election happens based on the interface link local address when the priorities are the same. The highest link local address switch is selected as the VRRP master.

A new master is elected when one of the following things happen:

- VRRP is disabled on the master router.
- Loss of communication occurs between master and backup router(s).

If VRRP is disabled on the master interface, the master router sends an advertisement with the priority set to 0 to all backup routers. This signals the backup routers that they do not need to wait for the master down interval to expire, and the master election process can begin immediately.

The master down interval is set using the following formula: 3 x advertisement interval + skew time.

The advertisement interval is a user-configurable option, and the skew time is (256-priority/256).

Note
The formula for VRRPv2; ((256 - priority) / 256). The VRRPv3 Skew time calculation will be different. Please refer to RFC 5798 (http://tools.ietf.org/html/rfc5798), Skew time. (((256 - priority) x Master_Adver_Interval) / 256)

Note
An extremely busy CPU can create a short dual master situation. To avoid this, increase the advertisement interval.

VRRP Master Preemption

VRRP master preemption is a feature that allows a VRRP backup router with a higher VRRP priority to take control from a lower priority backup that is acting as the master.

VRRP election occurs as described in ESRP Master Election on page 620. VRRP preemption occurs when a VRRP backup router is added to the network or recovers,

and that backup router has a higher priority than the current backup VRRP that is operating as the master.

Note
The VRRP router IP address owner always preempts, independent of the VRRP preemption setting.

When a VRRP backup router preempts the master, it does so in one of the following ways:

- If the preempt delay timer is configured for between 1 and 3600 seconds and the lower-priority master is still operating, the router preempts the master when the timer expires.
- If the preempt delay timer is configured for 0, the router preempts the master after three times length of the hello interval.
- If the higher priority router stops receiving advertisements from the current master for three times the length of the hello interval, it takes over mastership immediately.

The preempt delay timer provides time for a recovering router to complete start up before preempting a lower-priority router. If the preempt delay timer is configured too low, traffic is lost between the time the preempting router takes control and the time when it has completed startup.

VRRP Tracking

Tracking information is used to track various forms of connectivity from the *VRRP* router to the outside world.

VRRP Tracking Mode

When a *VRRP* tracked entity fails, the VRRP router behavior is controlled by the tracking mode. The mode can be all, or any. The default mode is all.

When the mode is all, the master role is relinquished when one of the following events occur:

- All of the tracked VLANs fail.
- All of the tracked routes fail.
- All of the tracked pings fail.

Note
Mastership is relinquished and the switch goes to INIT state.

When the mode is any, the master role is relinquished when any of the tracked VLANs, routes, or pings fail.

VRRP VLAN Tracking

You can configure *VRRP* to track active *VLAN*s (active ports in a VLAN or Loopback) of up to eight specified VLANs as criteria for failover.

If no active ports remain on the specified VLANs, the router automatically relinquishes master status based on the tracking mode.

When a tracking condition is in a failed state, VRRP behaves as though it is locally disabled; so it is neither master nor backup (which are both active states).

VRRP Route Table Tracking

You can configure *VRRP* to track specified routes in the route table as criteria for VRRP failover.

If any of the configured routes are not available within the route table, the router automatically relinquishes master status based on the tracking mode.

Note
MPLS (Multiprotocol Label Switching) LSPs are not considered to be part of VRRP route tracking.

VRRP Ping Tracking

You can configure *VRRP* to track connectivity using a simple ping to any outside responder.

The responder may represent the default route of the router, or any device meaningful to network connectivity of the master VRRP router. If pinging the responder consecutively fails the specified number of times, the router automatically relinquishes master status based on the tracking mode.

VRRP Address Support for IPv4

For IPv4 traffic, a primary IPv4 address is selected from the set of real interface addresses. *VRRP* advertisements are always sent using the primary IPv4 address as the source of the IPv4 packet.

The VRRP MAC address for an IPv4 VRRP router instance is an IEEE 802 MAC address in the following hexadecimal format (in Internet-standard bit-order):

```
00-00-5E-00-01-<vrid>
```

The first three octets are derived from the IANA Organizational Unique Identifier (OUI). The next two octets (00-01) indicate the address block assigned to the VRRP router for the IPv4 protocol, and VRID is the VRRP instance identifier. This mapping provides for up to 255 IPv4 VRRP routers on a network.

When a VRRP router instance becomes active, the master router issues a gratuitous ARP response that contains the VRRP router MAC address for each VRRP router IP address.

The master also always responds to ARP requests for VRRP router IP addresses with an ARP response containing the VRRP MAC address. Hosts on the network use the VRRP router MAC address when they send traffic to the default gateway.

VRRP Address Support for IPv6

IPv6 *VRRP* router advertisements are always sent using the VRRP virtual link-local address as the source address.

Hosts on the LAN can use this link-local address as their default route gateway. If no VRRP link-local address is configured, a default value is derived as follows:

```
FE80::5E00:02{VRID}
```

VRID is the VRRP instance identifier.

Note
The host portion of this address corresponds to the virtual MAC address associated with the VRRP router.

When a backup VRRP router assumes the role of master, it must use the same link-local address in router advertisements as the previous master. Therefore, when configuring a backup VRRP router, you must either configure a virtual link local address that matches the link local address on the IP address owner or allow the virtual link local address to default to the derived value in the same manner as on the master.

Router advertisement prefixes are configured based on the VRRP IP addresses. The mask used for the prefix will be the smallest mask used by all VRRP IP addresses on a *VLAN* interface. The ExtremeXOS software supports multiple IPv6 addresses on an interface that can overlap. For example, you can add both 200d::1/48 and 200d::2/96 to a VLAN. IPv6 VRRP routers advertise the smallest mask that applies to all VRRP IP addresses. In this example, if VRRP IP address 200d::100 is added, the mask is 48.

The VRRP MAC address for an IPv6 VRRP router instance is an IEEE 802 MAC address in the following hexadecimal format (in Internet-standard bit-order):

```
00-00-5E-00-02-<vrid>
```

The first three octets are derived from the IANA OUI. The next two octets (00-02) indicate the address block assigned to the VRRP router for IPv6 protocol, and VRID is the VRRP instance identifier. This mapping provides for up to 255 IPv6 VRRP routers on a network.

When a VRRP router assumes the role of master, it issues an unsolicited neighbor discovery (ND) neighbor advertisement message for each of the VRRP router IP addresses.

The master also always responds to ND neighbor solicitations with ND neighbor advertisements using the VRRP MAC address.

NTP VRRP Virtual IP support

This feature allows switches to configure the *VRRP* virtual IP as an NTP server address. The NTP server, when configured on the VRRP master, listens on the actual IP and virtual IP address for NTP clients.

On the VRRP backup node, only one socket is opened and bound to the physical IP address alone. Once a node transitions to the VRRP master, the ExtremeXOS software re-triggers a listen on the interface to ntpd for it to open a socket and bind to the VRRP VIP.

A flag configuration is added for the IPv4 cases, and these are propagated to *VLAN* Manager clients. NTP uses this to trigger a listen on the interface. For the master node to process non-ping packets destined to the VIP, the software already has a configuration command in VRRP (accept-mode on/off).

Note
If you want to configure VRRP VIP as the server address on NTP clients, enable accept mode.

Limitations

The following limitations exist for NTP VRRP Virtual IP support:

- ExtremeSwitching switches configured as NTP clients need to have the bootrom version: 2.0.1.7
- We do not recommend FHRP Virtual IPs for NTP configuration because they can cause undesirable behavior when the NTP servers are not in sync, or if the delay is asymmetric. Ensure that both servers derive their clock information from the same source.

  This problem can be more acute if a node connected to VRRP peers using and VRRP is in active-active mode. In this case, it is possible that every other packet could be sent to a different switch due to *LAG (Link Aggregation Group)* hashing at the remote node.

VRRPv3 Interoperation with VRRPv2

RFC 5798 states that VRRPv2 and VRRPv3 interoperation is optional, and that mixing these two versions should only be done when transitioning from VRRPv2 to VRRPv3.

VRRPv2 and VRRPv3 interoperation should not be implemented as a permanent solution.

This release supports configuration for the following modes of operation: VRRPv2, VRRPv3, and VRRPv2 and VRRPv3 interoperation.

VRRP Fabric Routing

Prior to ExtremeXOS 21.1, *VRRP* had one master router which would do L3 routing and one or more backup routers which would do L2 forwarding of packets to the master router. This caused loss of bandwidth in the links that connect the master and backup routers. This issue is present in any topology wherein host traffic is flowing via backup router. In case of multiple backup routers, traffic from hosts attached to some backup routers would have to traverse multiple links to reach the master router. This caused loss of bandwidth in multiple links towards the master.

The VRRP fabric routing feature introduced in ExtremeXOS 21.1 allows the backup router to take part in L3 routing for the packets it receives with DA=VMAC. A backup router enabled with this feature is called a Fabric Routing Enabled Backup (FREB) router. This feature allows load sharing of traffic between VRRP routers and saves the bandwidth of links connecting the master and backup. This bandwidth saved can be used to provide better connectivity between the segments of the network wherein each segment is connected to only one VRRP router directly. This solution is applicable for all topologies such as , *EAPS (Extreme Automatic Protection Switching)*, or *STP (Spanning Tree Protocol)*.

The VRRP FREB router receives and processes VRRP advertisements from the master, but does not generate advertisements. Broadcast ARP requests and multicast neighbor solicitations to resolve the virtual IP are responded to by the master router. The Fabric Routing Enabled Backup router behaves as mentioned in RFC 5798 with the following exceptions:

- The backup router performs forwarding functionality, similar to the master.
- The backup router responds to unicast ARP/Neighbor Solicitation requests it receives.

Configuring VRRP Fabric Routing

To configure the fabric routing feature, enter the following command:

```
configure vrrp vlan vlan_name vrid vridval fabric-routing [on | off]
```

This configuration can be present on all VRRP routers, regardless of VRRP state of the router. Fabric routing will be enabled only when the VRRP router is in backup state.

Fabric Routing Functionality

The *VRRP* backup router enabled with fabric routing mode is called a VRRP FREB router (Fabric Routing Enabled Backup router). The VRRP FREB router is responsible for the following:

- Performing routing for the packets destined for subnets other than the received interface's subnet. These packets should also have a destination MAC matching Virtual MAC. These packets will be routed by hardware.
- Forwarding the packets having a Destination IP matching Virtual IP, towards VRRP Master. These packets will have Destination MAC matching Virtual MAC. These packets will be forwarded by hardware.
- Responding for unicast ARP requests with target IP matching Virtual IP.
- Responding for unicast Neighbor Solicitation requests with target IP matching Virtual IP.
- Does not respond to broadcast ARP / multicast Neighbor solicitations targeted for Virtual IP.
- Does not respond for *ICMP (Internet Control Message Protocol)* requests destined for Virtual IP.
- Does not generate gratuitous ARP/ Neighbor Advertisements.
- Does not advertise Router Advertisement Prefixes in this state.

VRRP Router Accepts Packets Destined to Virtual IP

Only the master serves as protocol servers like NTP, telnet, SSH, etc and accepts connections destined to the virtual IP. By doing so, hosts always connect to the same by using the virtual IP, at any given point of time. This ensures that the host is getting a consistent response from the protocol server. This arrangement allows the host to reach the NTP server using the same IP i.e. virtual IP, even if VRRP mastership moves to a different router. A network monitoring tool is another example, which can use virtual IP to collect data about VRRP domain, by connecting to current VRRP Master. It is not recommended to change the configurations of the switch, when a management session is connected using virtual IP. When VRRP FREB router sits in between the host and VRRP Master router, FREB router does hardware forwarding of these packets from host towards VRRP Master, at Layer 3.

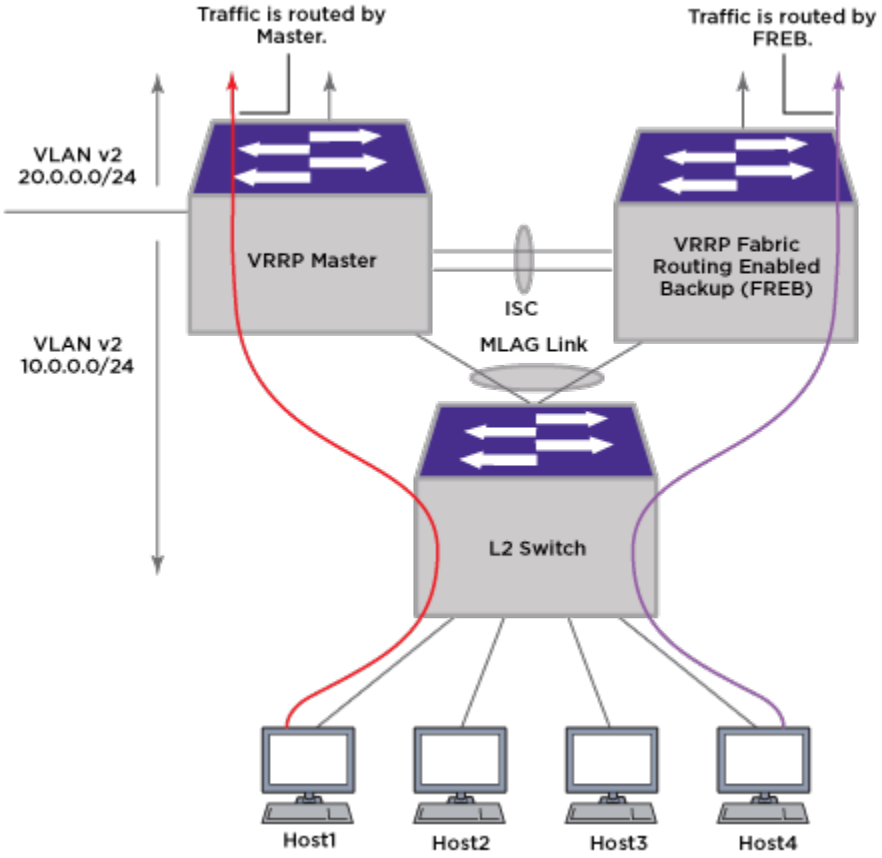


Figure 261: VRRP Fabric Routing

Note
A caveat is that TTL/hop count is decremented for the packets destined for virtual IP, when forwarded by FREB. This may be a problem to run any protocol that expects TTL not to be decremented, between host and Master.

Hosts can generate unicast ARP to validate a ARP cache entry. Similarly, unicast Neighbor Solicitation is generated to perform Neighbor Unreachability Detection for a neighbor. These requests are periodic. The unicast ARP/NS requests will be responded by FREB, if it receives the request. A downside of allowing VRRP Master to respond

these requests is that it may take considerable CPU cycles when large numbers of hosts are present in VRRP domain.

Enabling Fabric Routing Mode

A *VRRP* router configured with fabric routing mode will be in backup state. There is some difference in behavior based on enabling sequence which is described below.

When enabling VRRP VR after configuring fabric routing mode, assume that fabric routing mode has already been configured for VRRP VR and VR is administratively enabled now. Initially the VRRP state machine will start from INIT state. It will move to backup state and wait for Master_Down_Interval duration to receive an advertisement from current master. Based on VRRP Master election (as described earlier in the sections), VRRP router will either become master or will assume forwarding responsibility while in backup state (i.e., FREB). Waiting in backup state avoids a burst of messages exchanged in software during the transition. This is useful in scaled setups.

Note
For the duration it waits in backup state, traffic will be forwarded to the VRRP master where it gets routed. This behavior is same as existing behavior without fabric routing.

When configuring fabric route mode when the VR is already in backup state, if the VR has received advertisements within the last advertisement interval, the router will take forwarding responsibility immediately.

The hold off timer will be applied before FREB starts IP forwarding. In case of reboot or process restart, the VR will wait for 60 seconds before taking forwarding responsibility. This will allow the routing protocols in the restarting system to converge.

To enable fabric routing, use the following command:

`configure vrrp` **`vlan`**`[ vlan_name | vlan_list] [vridval |` **`all`**`} {`**`group`** `group_name}`**`fabric-routing`** `[`**`on`** `|` **`off`**`]`

To configure a specific VRRP instance as FREB:

```
configure vrrp vlan v1 vrid 1 fabric-routing on
```

To configure all the VRRP instances as FREB:

```
configure vrrp all fabric-routing on
```

Configured fabric routing can be seen in the show command output of the `show vrrp detail` or `show vrrp vlan vlan` CLI command:

```
# sh vrrp detail
VLAN:  v1       VRID:  1        VRRP:  Disabled State:  Backup
Virtual Router:  VR-Default
Priority:  100(backup)  Advertisement Interval:  1 sec
Version: v2     Preempt:  Yes   Preempt Delay: 0 sec
 Virtual IP Addresses:
 Accept mode:  Off
```

```
Host-Mobility:  Off
Host-Mobility Exclude-Ports:
Tracking mode:  ALL
Tracked Pings:  -
Tracked IP Routes:  -
Tracked VLANs:  -
Fabric Routing: On

 * indicates a tracking condition has failed
```

Fabric Routing Limitations

The following are limitation of *VRRP* fabric routing:

- VRRP fabric routing is not supported for the configuration of virtual IP address being same as interface IP address (i.e., IP owner case).
- VRRP fabric routing is not supported for PVLAN and VLAN aggregation topologies.

VRRP and Hitless Failover

Note
This section applies to SummitStack only.

The primary node executes the switch's management functions, and the backup acts in a standby role. Hitless failover transfers switch management control from the primary to the backup and maintains the state of *VRRP*. While VRRP supports hitless failover, you do not explicitly configure hitless failover support; rather, if you have two nodes installed, hitless failover is available.

To support hitless failover, the primary node replicates VRRP protocol data units (PDUs) to the backup, which allows the nodes to run VRRP in parallel. Although both nodes receive VRRP PDUs, only the primary transmits VRRP PDUs to neighboring switches and participates in VRRP.

To initiate hitless failover on a network that uses VRRP:

1. Confirm that the primary and backup nodes are synchronized and have identical software and switch configurations using the `show switch {detail}` command.
   The output displays the status of the nodes, with the primary node showing MASTER and the backup node showing BACKUP (InSync).
   a. If the primary and backup nodes are not synchronized and both nodes are running a version of ExtremeXOS that supports synchronization, proceed to step 2 in STP and Hitless Failover—SummitStack Only on page 1808.
   b. If the primary and backup nodes are synchronized, proceed to step 3 in STP and Hitless Failover—SummitStack Only on page 1808.
2. If the primary and backup nodes are not synchronized, use the `synchronize` command to replicate all saved images and configurations from the primary to the backup.
   After you confirm the primary and backup nodes are synchronized, proceed to step 3 in STP and Hitless Failover—SummitStack Only on page 1808.
3. If the primary and backup nodes are synchronized, use the `run failover` (formerly `run msm-failover`) command to initiate failover.

For more detailed information about verifying the status of the nodes and system redundancy, see Understanding System Redundancy on page 84. For more information about hitless failover, see Understanding Hitless Failover Support on page 89.

Note
For complete information about software licensing, including how to obtain and upgrade your license and what licenses are appropriate for these features, in the Switch Engine v33.6.1 Licensing Guide document.

VRRP Host Mobility

VRRP host mobility solves the asymmetric routing problem associated with VRRP where the path to return to an end host may be different and longer than necessary.This feature uses host routes to indicate where in the network an end host resides. Using other routing protocols such as *OSPF (Open Shortest Path First)*, other routers will them pick the shortest path back to the end host when multiple paths are available via *ECMP (Equal Cost Multi Paths)* route entries.

Asymmetric Routing and VRRP

The optimum flow path for frames sourced by clients is that they are forwarded by the first encountered router in the path. The optimum flow path for return frames for these flows is through the same router that forwarded the initial frames of the flow. This is referred to as a symmetric routing path. Unfortunately, standard VRRP elects one of a group of routers to forward frames from a client, but the return path may be through any of the routers in the VRRP group. This condition is known as asymmetric routing. In the following figure there are three devices that are routers in a VRRP group servicing the 192.168.1.0/24 network with the VRRP virtual gateway 192.168.1.1. Router R2 has been elected VRRP master and this is the only router that will route frames from the clients to the core, even though the clients are not locally connected to the master. The other devices bridge frames from the clients to the master; therefore, the frames from User55 and User77 must traverse two devices when routing the frame to the core.

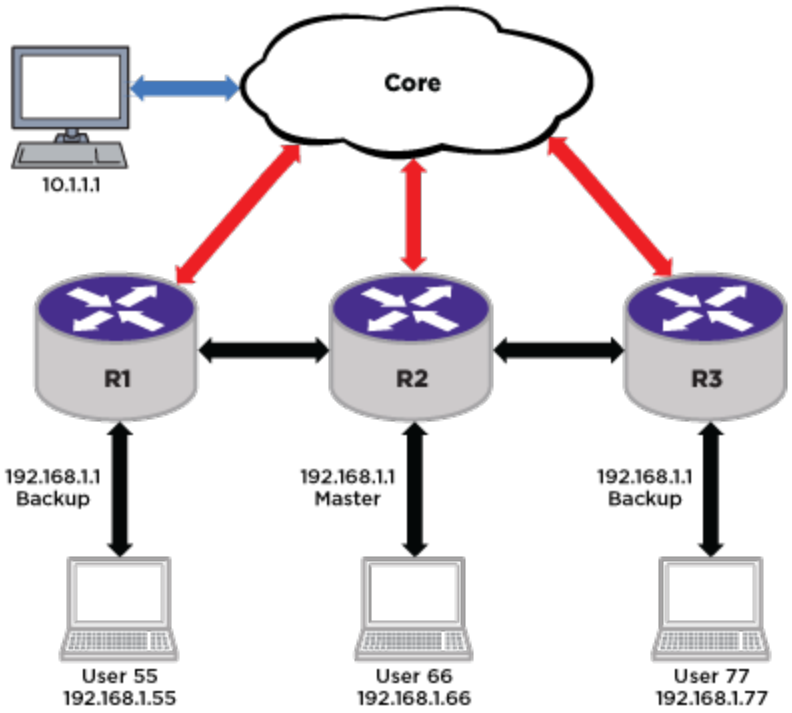


Figure 262: Asymmetric Routing and VRRP (1)

R1, R2, and R3 all advertise 192.168.1.0/24 to the core. This means that return traffic to the users will be sent from the core to any of these routers. A non-optimal path for the flows in either direction may result. Flows from User 55 to Core would traverse R1, R2. The return path could traverse R3, R2, and R1.

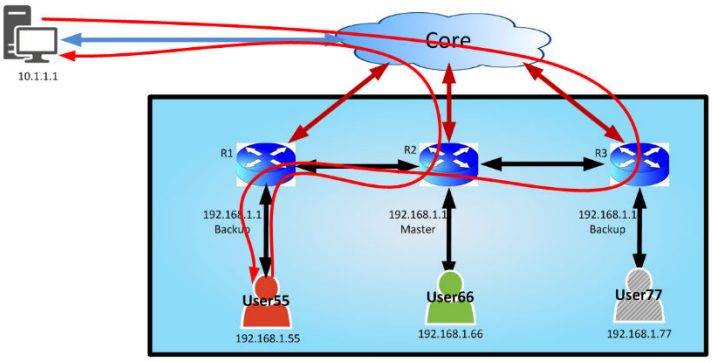


Figure 263: Asymmetric Routing and VRRP (2)

Fabric Routing solves part of this path problem by allowing the backup routers to forward frames for the clients without having to send the frame to the master router for the group. Now frames from User 55 will be routed by the first router that receives the frame (R1). Unfortunately, the return path to User 55 from the core is still via Equal Cost Multipath (ECMP) to any of the routers. Again, the frame may have to still traverse R3, R2, and finally R1 to reach User 55 from the core.

Host Mobility Solving Asymmetric Routing

The host mobility feature uses ARP/ND to learn each user connected to the IP Network on the router that is closest to the end host. It then uses routing protocols such as OSPF to advertise the hosts IP/IPv6 address. These “Host Routes” will be advertised to other routers making them the preferred route to the end host’s IP address. As the host route propagates through the network the most efficient path to the end host will be revealed as every other path will have an additional cost. With host mobility, User 55’s complete route will be advertised to R2, R3 and the core. The core will then receive a route to User 55’s address from all three routers. The most direct route will have less hops and a lower cost than the routes reported by R2 and R3. When traffic destined to User 55 enters the core the most direct path (R1) is selected.

VRRP Host Mobility Feature Detail

The ExtremeXOS Host-Mobility feature is implemented using the user space application VRRP, which listens for ARP/ND updates from the kernel using the *FDB (forwarding database)*. When ARP/ND entries are inserted into the kernel, the FDB notifies the host-mobility application. If the new ARP/ND entry exists on an interface that has VRRP, host-mobility is configured on the VRRP & VRID pair, and the port is not being excluded, the host-mobility application will create a host route in the route table.

The host routes in the route table will be propagated via existing routing protocols such as OSPF.

As ARP/ND entries are removed from the kernel, the host-mobility portion of the VRRP application is notified. Normally if the interface is still up when entries are removed, the host-mobility application sends an ARP/ND packet to the IP address of the host associated with the ARP/ND entry to resolve the address again. The FDB attempts to resolve ARP addresses when near the age period. Due to the FDB keeping ARPs up to date, host-mobility does not need to try to resolve aging entries when aging. When the entry is removed by a clear CLI command, FDB does not send an ARP. When the entry is removed, Host-Mobility requests an ARP. After 5 seconds, if no route has been added, the host routes are removed from the route table.

Ports are designated as host or router ports; by default the ports are host ports. ARP/ND entries learned on router ports do not generate host route entries. Only ARP/ND entries learned on host ports generate route entries. Host route entries are removed if the ARP/ND entry is learned or changes to a router port.

When a host moves from one switch to another on the same layer 2 network, a new host route is created from the new location. In this case there may be two host routes throughout the network advertising for the same host. While not optimal, this situation does not cause any network delivery problems. Eventually the original host route is removed when the ARP/ND entry is removed from the switch or moves from a host port to a router port.

A redistribution command is required for OSPF or other routing protocols to distribute the new host routes. The command to redistribute is “enable ospf export host-mobility cost 0 type ase-type-2”. A new routing “Origin” type is added to route table for host-mobility routes.

Since OSPF will redistribute host-mobility routes, it is possible that a route could be learned for a host that the local device can reach at a lesser cost. Host-mobility monitors routes added by OSPF. If OSPF adds a route that is part of a subnet that the device is a member of, then host-mobility triggers ARP/ND to be performed. By performing ARP/ND, the lower cost route to host is added as an ARP/ND entry in the proper table and is used while routing traffic. If no response is received, then the OSPF route will continue to be used instead.

VRRP Grouping to Increase VR Scaling

To achieve scaling of *VRRP* VR instances above 511 to as high as 2,048, selected VRRP VRs must be grouped as instructed by the user. Groups involving the same set of VRRP VRs are configured in all VRRP routers. A single chosen VR out of the group (primary VR) transmits/receives VRRP advertisements at the configured interval. Other VRs (secondary VRs) in the group transmit/receive at a much slower rate. This requires a much fewer number of VRRP advertisements exchanged, and reduces CPU resource usage, increasing scaling. All VRs in a group assume same VRRP state in a given router.

VRRP Fabric Routing needs to be enabled on all member VRs of a group in participating VRRP routers. This does not have any effect when a VR (or group of VRs) is master; but, when a higher priority master becomes available, this VR transitions to Fabric Routing-enabled backup, instead of regular backup. Since the Fabric Routing feature enables VRRP backup routers to load share by Layer 3 forwarding, and conforms to RFC 5798, this mode is preferred over using a regular backup router. However, if you want the regular master-backup configuration, you can use a separate VRID that is not used by VRs in any group.

VRRP Grouping Limitations

- All *VRRP* VLANs participating in same group should have same topology in a given router.
- VRRP groups and individual VRs can not be mixed when VRs use the same VRID and are configured to operate in regular master-backup mode.
- Fabric Routing must be enabled for the group of VRs if the same VRID is repeated in multiple groups.
- When the number of VRs configured on a router exceeds 511, you must switch to high-scale mode (use VRRP grouping) to obtain scaling to 2,048 VR instances.
- The same group does not support combinations of some IPv4 VRs and some IPv6 VRs. Separate groups must be used for each type.

VRRP Grouping Description

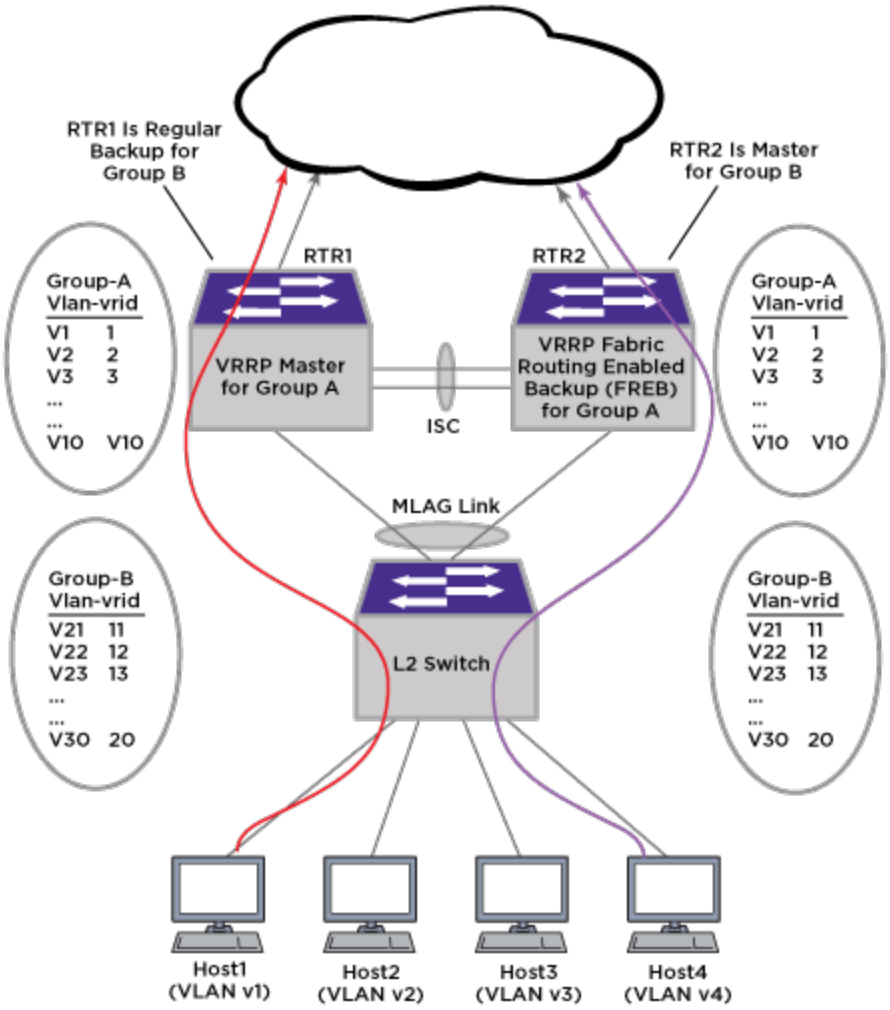


Figure 264: Example VRRP Group Configuration

Figure 264 shows an example configuration of two different groups in simple topology. Group A is configured with 10 VRRP VRs, each VR on a separate *VLAN*, and using separate VRID. The same set of *VRRP* VLANs are grouped together on both routers. All the VRRP VLANs on Group A have the same topology with respect to Router 1 and Router 2. If a VLAN on Router 1 goes down due to link faults or I/O slot failure, other VLANs also go down (that is, all VLANs in the group share the same fate).

This is a typical topology for which this scaling is well suited. If the VLAN going down event triggers VRRP state change in one VLAN, a VRRP state change occurs on other VLANs also. This example has four hosts on different VLANs belonging to group A; traffic from hosts may get hashed to either of the MLAG links and reach either the VRRP master (RTR1) or the Fabric Routing-enabled backup router (RTR2). Both routers perform routing for host traffic. Group B has a different set of 10 VRRP VRs that were configured on a different set of VLANs and with a different set of VRIDs from that

of Group A. Routers are configured for regular master-backup states (unlike Group A which has master-FREB states).

Note
If you want a regular master-backup operation on a particular group for some reason, do not reuse the VRIDs configured in other groups or other individual VRs.

Enabling Fabric Routing on VRs

Enabling Fabric Routing is only required if a VRID is reused in more than one group. Only these groups of VRs need to be enabled for Fabric Routing in all participating *VRRP* routers.

Grouping Reusing VRIDs in Different Groups (Fabric Routing Mode)

Table 177 shows an example configuration that reuses VRIDs in multiple groups. All three groups are configured on each VRRP router. Notice that Groups A and C have the same set of *VLAN*s, but VRID is interchanged between the VLANs. Since all three groups are sharing the same VRIDs, Fabric Routing needs to be enabled on VRs in all three groups.

Table 177: VRRP Grouping Example #1

Group A		Group B		Group C	
VLAN	VRRP VRID	VLAN	VRRP VRID	VLAN	VRRP VRID
V1	1	V10	1	V1	4
V2	2	V11	2	V2	3
V3	3	V12	3	V3	2
V4	4	V13	4	V4	1

Grouping with Unique VRIDs

All VRs in Group I use VRID 1 in Table 178. All VRs in Group J use VRID 2. In this configuration, all VR groups do not need to be enabled for Fabric Routing. Group I can operate in master-FREB mode, and Group J can choose to operate in master-backup mode. In this case, only VRs in Group I need to be enabled for Fabric Routing.

Table 178: VRRP Grouping Example #2

Group I		Group J		Group K	
VLAN	VRRP VRID	VLAN	VRRP VRID	VLAN	VRRP VRID
V1	1	V10	2	V1	3
V2	1	V11	2	V2	3
V3	1	V12	2	V3	3
V4	1	V13	2	V4	3

Grouping with Multiple VRIDs per Group with VRIDs Not Repeated

It is possible to have an individual VR when there are some groups already in the same router. This may be useful when this router has two VRRP peer routers and it has to operate in group mode with one peer, but has to operate in individual VR mode with another peer at the same time. In that case, an individual VR must be configured with a unique VRID that is not used in any of the groups or other individual VRs. This unique VRID is required only if either a group or individual VR is operating in master-backup mode. However, if all groups and individual VRs are operating in master-FREB mode, then VRIDs used within a group can be assigned to individual VRs.

Table 179: VRRP Grouping Example #3

Group L		Group M		Group N	
VLAN	VRRP VRID	VLAN	VRRP VRID	VLAN	VRRP VRID
V1 V2 V3 V4	1 2 3 4	V10 V11 V12 V13	5 6 7 8	V1 V2 V3 V4	9 10 11 12

Characteristics of VRs in a Group

When VRs are grouped together you need to choose a VR from the group and configure it as the primary VR using the CLI. Other VRs belonging to the group need to be added as secondary VRs. When a group is operational, the following approach maintains the same state across all VRs:

- The primary VR sends *VRRP* advertisements at the configured interval.
- All the VRs in the group have the same VRRP state in a given router.
- Only the primary VR registers for tracking objects (iproute tracking, ping tracking, *VLAN* tracking) and processes tracked events. If any tracking configuration is present on secondary VRs, they are not effective when the VR is part of a group.
- Secondary VRs inherit the priority of the primary VR. This is useful because when you want to change the priority of a group, you only need to change priority of the single primary VR.
- The pre-empt configuration of the primary VR is applied for the group. If these configurations (pre-empt allowed, pre-empt delay) are present in secondary VRs, they are not effective.
- The advertisement interval configuration on a secondary VR is also not effective.

Other per VR configurations/properties should continue to work as-is, such as accept-mode and host-mobility. All secondary VRs send advertisements at 40 seconds intervals. Sending advertisements help Layer 2 switches in the network to learn VMACs on a secondary VR's VLAN. Sending at a slower rate reduces processing load at control plane, on VRRP peers.

The other properties of individual VRs present in a group are not altered from their current behavior.

Switching Between Individual VR Mode and High-Scale Mode (Group)

It may be preferable to run in individual VR mode until VR count exceeds its maximum limit (511 instances), as it gives more flexibility with respect to topology choices and 'per-*VLAN* routing control for the same VRID'. It will allow the VRs configured in different VLANs configured with same VRID to be in different routing states (for example, one VR can be master and that VLAN can do routing, but the other VR can be backup and that VLAN can do Layer 2 forwarding towards the master).

When you try to configure and enable a VR beyond 511 VRs, you are notified to configure *VRRP* groups and add VRs to appropriate groups. VRs configured on VLANs following the same topology can be grouped together. There can be multiple groups for the same topology for administrative reasons, or for running with different VRRP configurations (like advertisement interval, etc.), or with different Fabric Routing mode.

Alternatively, groups can be configured beforehand and kept for future use. Similarly, VRs can be added to the group beforehand. The groups can be kept in a disabled state (`disable vrrp` **`group`** *`group_name`* {**`configuration`** | **`members`**}) until you are notified to configure VRRP groups (VR count exceeds 511). When groups are in the disabled state, the VRs present in the group configuration run in individual VR mode.

During switchover, it is recommended to execute the commands:

`disable` **`vrrp`**

`disable vrrp` **`group`** *`group_name`* **`members`**

After adding VRs to group, and enabling group configuration, execute the commands:

`enable` **`vrrp`**

`enable vrrp` **`group`** *`group_name`* **`members`**

Keeping VRs in disabled state during transition, and then enabling afterwards, ensures that the bulk transfer of VR information across the system occurs more easily.

You can switch to individual VR mode from high-scale mode by disabling all VRRP groups. Since the limit of VRs in individual mode is 511, ensure that greater than 511 VRs are not enabled before disabling the group configuration.

When a group is already operational (the group is enabled, and also its member VRs are enabled), additional VR can be added to the group. It is recommended to disable the VR before adding it to a group that is currently running and forwarding traffic. Enable the VR after it is added to group in all participating routers. This avoids traffic loss during the transition to group mode.

You can have few individual VRs outside of groups. If you are transitioning to high-scale mode when an individual VR is present, disable all VRs (`disable vrrp`) before the transition and enable all VRs (`enable vrrp`) after the transition.

For a list of all VRRP group commands, see Configuring VRRP Groups on page 2300.

Note
If a large number of VRs are added under the same group, the advertisement interval of the primary VR of the group should be increased to multiples of seconds. This reduces the processing load on the VRRP backup router and avoids protocol flaps. For a large IPv6 group, chose an advertisement interval of a greater value than the value of a similarly sized IPv4 group.

Configuring VRRP

The following procedure can be used to configure a simple *VRRP* topology:

1. Configure the VRRP IP address owner router as follows:
   a. Create a *VLAN* to serve as the VRRP router VLAN.
   b. Add an IP address to the VRRP VLAN.
   c. Enable IP forwarding on VRRP VLAN.
   d. Add the VRRP version for the VRRP instance (see Configuring VRRP Version Support on page 2299).
   e. Create the VRRP router instance for the intended VRRP master (see Creating and Deleting VRRP Router Instances on page 2298).
   f. Add the IP address defined in Step 1b as a VRRP router IP address (see Adding and Deleting VRRP Router IP Addresses on page 2298).
   g. Enable VRRP on the switch (see Enabling and Disabling VRRP and VRRP Router Instances on page 2301).
2. Configure each backup VRRP router as follows:
   a. Create a VLAN to serve as the VRRP router VLAN. This name must match the name used for the appropriate VRRP IP address owner.
   b. Add an IP address to the VRRP VLAN. This address must be different from the IP address assigned to the intended VRRP master, but it must use the same subnet.
   c. Enable IP forwarding on VRRP VLAN.
   d. Create the VRRP router instance that will serve as the backup instance (see Creating and Deleting VRRP Router Instances on page 2298).
   e. Configure the priority for the backup VRRP router to a value in the range of 1–254 (see Configuring VRRP Router Priority on page 2299).
   f. Add the same VRRP router IP address that was added to the intended VRRP master instance (see Adding and Deleting VRRP Router IP Addresses on page 2298).
   g. Enable VRRP on the switch (see Enabling and Disabling VRRP and VRRP Router Instances on page 2301).
3. Configure network workstations to use the VRRP router IP address as the default gateway address.

Creating and Deleting VRRP Router Instances

- To create a *VRRP* router instance, use the following command:
  `create vrrp vlan vlan_name vrid vridval`
- To delete a VRRP router instance, use the following command:
  `delete vrrp vlan vlan_name vrid vridval`

Adding and Deleting VRRP Router IP Addresses

- To add a *VRRP* router IP address to a switch, use the following command:
  `configure vrrp vlan vlan_name vrid vridval add ipaddress`

  Note
  A VRRP routing instance can support IPv4 or IPv6 addresses, but it cannot support both.

- To delete a VRRP router IP address from a switch, use the following command:
  `configure vrrp vlan vlan_name vrid vridval delete ipaddress`

Adding an IPv6 Link Local Address to a VRRP Router

To add an IPv6 link local address to a *VRRP* router, enter the following command:

`configure vrrp vlan vlan_name vrid vridval add virtual-link-local vll_addr`

Configuring the VRRP Router Advertisement Interval

To configure the *VRRP* router advertisement interval, use the following command:

`configure vrrp vlan vlan_name vrid vridval advertisement-interval interval [{seconds} | centiseconds]`

Note
We recommend that you configure the same router advertisement interval in all VRRP routers. VRRPv3 supports a 40 second maximum advertisement interval.

Note
If the advertisement interval is different on the master and backup switches, the master switch's advertisement interval is used by the backup as well.

Configuring VRRP Router Advertisement by Link Local Address Only

To configure the router discovery advertisements to send only with VRRP link local address on the VRRP-enabled VLAN interface, use the following command:

`configure {vlan} vlan_name router-discovery {ipv6} vrrp-lla-only on_off`

Configuring Master Preemption

- To enable *VRRP* master preemption and configure the preempt delay timer, use the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval preempt {delay seconds}
  ```

- To disable VRRP master preemption, use the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval dont-preempt
  ```

Configuring VRRP Router Priority

To configure the priority for a *VRRP* router, use the following command:

```
configure vrrp vlan vlan_name vrid vridval priority priorityval
```

Note
If VRRPv3 and VRRPv2 routers participate in VRRP instance, the VRRPv3 routers should be configured with a higher priority to ensure that they win master elections over VRRPv2 routers.

Configuring the Accept Mode

To configure the accept mode, enter the following command:

```
configure vrrp vlan vlan_name vrid vridval accept-mode [on | off]
```

Configuring NTP VRRP Virtual IP support

To configure the *VRRP* virtual IP as NTP server address, enter the following command:

```
configure vrrp vlan vlan_name vrid vridval accept mode [on|off]
```

Enabling the accept mode allows the switch to process non-ping packets that have a destination IP set to the virtual IP address.

Configuring VRRP Version Support

To configure *VRRP* version support, use the following command:

```
configure vrrp vlan vlan_name vrid vridval version [v3-v2 | v3 | v2]
```

Configuring VRRP Tracking

Configuring the Tracking Mode

To configure the tracking mode, enter the following command:

```
configure vrrp vlan vlan_name vrid vridval track-mode [all | any]
```

Adding and Deleting Tracked Routes

- To add a tracked route, enter the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval add track-iproute
  ipaddress/masklength
  ```

- To delete a tracked route, enter the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval delete track-iproute
  ipaddress/masklength
  ```

Adding and Deleting Tracked VLANs

- To add a tracked *VLAN*, enter the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval add track-vlan
  target_vlan_name
  ```

- To delete a tracked VLAN, enter the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval delete track-vlan
  target_vlan_name
  ```

Adding and Deleting Tracked Pings

- To add a tracked ping, enter the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval add track-ping ipaddress
  frequency seconds miss misses {success successes}
  ```

- To delete a tracked ping, enter the following command:

  ```
  configure vrrp vlan vlan_name vrid vridval delete track-ping ipaddress
  ```

Configuring VRRP Fabric Routing

To configure the fabric routing feature, enter the following command:

```
configure vrrp vlan[ vlan_name | vlan_list] [vridval | all} {group
group_name}fabric-routing [on | off]
```

Configuring Host Mobility

To configure host-mobility, enter the following command:

```
configure vrrp [{vlan} [vlan_name | vlan_list] vrid [vridval | vrid_list
| all]] host-mobility [{on | off} {exclude-ports [add | delete]
port_list}]
```

In general, the port that is not connected to host is added in exclude-ports to avoid unnecessary host route.

Configuring VRRP Groups

To create *VRRP* groups:

```
create vrrp group group_name
```

To delete VRRP groups:

```
delete vrrp group group_name
```

To add/delete primary VRs or secondary VRs:

```
configure vrrp group group_name add [primary-vr | secondary-vr] [{vlan}
vlan_name vrid vridval | vlan vlan_list {vrid vrid_list}]
```

```
configure vrrp group group_name delete [primary-vr | secondary-vr
[{vlan} vlan_name vrid vridval | vlan vlan_list {vrid vrid_list} |
all] ]
```

To enable VRRP groups:

```
enable vrrp group group_name {configuration | members}
```

To disable VRRP groups:

```
disable vrrp group group_name {configuration | members}
```

To show VRRP group information:

```
show vrrp group group_name
```

For an example VRRP group configuration, see VRRP Groups Configuration Example on page 2303.

Managing VRRP

Enabling and Disabling VRRP and VRRP Router Instances

- To enable *VRRP* or a VRRP router instance, enter the following command:
  ```
  enable vrrp {vlan vlan_name vrid vridval}
  ```
- To disable VRRP or a VRRP router instance, enter the following command:
  ```
  disable vrrp {vlan vlan_name vrid vridval}
  ```

Clearing VRRP Counters

To clear the *VRRP* counters, enter the following command:

```
clear counters vrrp {{vlan vlan_name} {vrid vridval}}
```

Displaying VRRP Information

Displaying VRRP Router Information

To display *VRRP* router information for one or all VRs, enter the following command:

```
show vrrp {virtual-router {vr_name}} {detail}
```

Displaying VRRP Router Information and Statistics for VLANs

To display *VRRP* information or statistics for a *VLAN*, enter the following command:

- Display VRRP information or statistics for a VLAN.

  ```
  show vrrp vlan vlan_name | vlan_list {stats}
  ```

Displaying VRRP Tracking Information

To view the status of tracked devices, enter the following command:

```
show vrrp {virtual-router {vr_name}} {detail}
```

VRRP Configuration Examples

Simple VRRP Network Examples

The topology for a simple *VRRP* network example is shown in Figure 259 on page 2277.

Switch A is the IP address owner, and Switch B is configured as the backup.

The configuration commands for switch A are as follows:

```
configure vlan vlan1 ipaddress 192.168.1.3/24
enable ipforwarding vlan vlan1
create vrrp vlan vlan1 vrid 1
configure vrrp vlan vlan1 vrid 1 priority 255
configure vrrp vlan vlan1 vrid 1 add 192.168.1.3
enable vrrp
```

The configuration commands for switch B are as follows:

```
configure vlan vlan1 ipaddress 192.168.1.5/24
enable ipforwarding vlan vlan1
create vrrp vlan vlan1 vrid 1
configure vrrp vlan vlan1 vrid 1 add 192.168.1.3
enable vrrp
```

VRRP Load Sharing Example

You can use two or more *VRRP*-enabled switches to provide a fully redundant VRRP configuration that supports load sharing on your network.

The topology for a load sharing example is shown in Figure 260 on page 2278. Switch A is the IP address owner for VRRP router instance 1 and the backup for VRRP instance 2. Switch B is the IP address owner for VRRP router instance 2 and the backup for VRRP instance 1.

The configuration commands for switch A are as follows:

```
configure vlan vlan1 ipaddress 192.168.1.3/24
enable ipforwarding vlan vlan1
```

```
create vrrp vlan vlan1 vrid 1
configure vrrp vlan vlan1 vrid 1 priority 255
configure vrrp vlan vlan1 vrid 1 add 192.168.1.3
create vrrp vlan vlan1 vrid 2
configure vrrp vlan vlan1 vrid 2 add 192.168.1.5
enable vrrp
```

The configuration commands for switch B are as follows:

```
configure vlan vlan1 ipaddress 192.168.1.5/24
enable ipforwarding vlan vlan1
create vrrp vlan vlan1 vrid 2
configure vrrp vlan vlan1 vrid 2 priority 255
configure vrrp vlan vlan1 vrid 2 add 192.168.1.5
create vrrp vlan vlan1 vrid 1
configure vrrp vlan vlan1 vrid 1 add 192.168.1.3
enable vrrp
```

VRRP Tracking

Tracking information is used to track various forms of connectivity from the *VRRP* router to the outside world.

VRRP Groups Configuration Example

Note

This configuration example assumes that VRs are already created and configured with regular *VRRP* parameters, but not yet enabled (`enable vrrp {vlan [vlan_name | vlan_list] vrid [vridval | vrid_list}` command is not yet executed). If VRs are already enabled, disable all the VRs (`disable vrrp {vlan [vlan_name | vlan_list] vrid [vridval | vrid_list]}`).

1. Create VRRP group “groupA”:
   ```
   create vrrp group “groupA”
   ```
2. Add primary VR before adding any secondary VRs to the group:
   ```
   configure vrrp group “groupA” add primary-vr vlan “vlan1” vrid 1
   ```
3. Add secondary VRs one at a time:
   ```
   configure vrrp group “groupA” add secondary-vr vlan “vlan2” vrid 2
   configure vrrp group “groupA” add secondary-vr vlan “vlan3” vrid 3
   configure vrrp group “groupA” add secondary-vr vlan “vlan4” vrid 4
   configure vrrp group “groupA” add secondary-vr vlan “vlan5” vrid 5
   ```

   Or alternatively, *VLAN* tag range and VRID range can be specified. In this case, vlan1 has vlan-id 101, vlan2 has vlan-id 102, etc.:

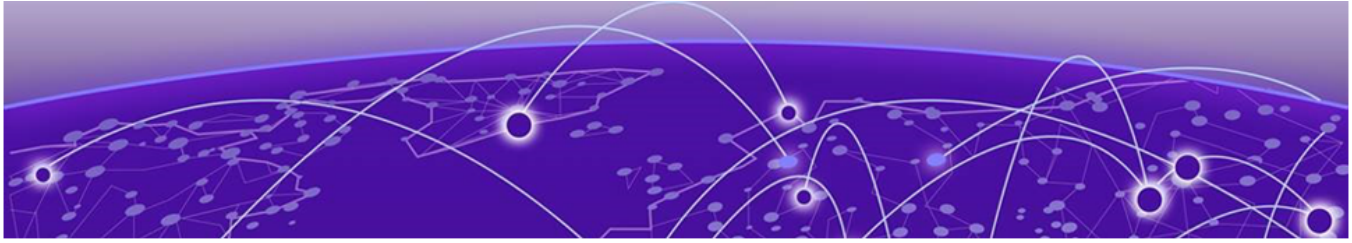
   ```
   configure vrrp group “groupA” add secondary-vr vlan 102-105 vrid 2-5
   ```
4. Configure fabric routing on all the member VRs in the given group:
   ```
   configure vrrp group "groupA" fabric-routing on
   ```

5. Apply group configuration on member VRs. Before enabling the group, member VRs are operating in individual VR mode:
   ```
   enable vrrp group groupA
   ```
6. Enable all VRs using the global command, if there is only one group:
   ```
   enable vrrp
   ```

   or enable only member VRs of this group:
   ```
   enable vrrp group groupA members
   ```

After configuring VRRP groups, the following appears for the below show commands:

```
# show vrrp group
                                                Members
                  Group Name    Priority     State Count    TP/TR/TV/P/T   /FR
                  groupA(En)       100       MSTR  6         0  0  0 Y 1     Y

  En-Enabled, Ds-Disabled, Pri-Priority, T-Advert Timer, P-Preempt
  TP-Tracked Pings, TR-Tracked Routes, TV-Tracked VLANs, FR-Fabric Routing,
  MX-Mixed values

# show vrrp vlan1
VLAN:  vlan1    VRID:  1        VRRP:  Enabled  State:  MASTER
Virtual Router:  VR-Default
Priority:  100(backup)  Advertisement Interval:  1 sec
Version: v3-v2  Preempt:  Yes   Preempt Delay: 0 sec
 Virtual IP Addresses:
 Accept mode:  Off            Group : groupA (Enabled)
 Host-Mobility:  Off
 Host-Mobility Exclude-Ports:
 Checksum:  Include pseudo-header
 Tracking mode:  ALL
 Tracked Pings:  -
 Tracked IP Routes:  -
 Tracked VLANs:  -
 Fabric Routing: On

  * indicates a tracking condition has failed
```



VXLAN

VXLAN Overview on page 2306
Overlay Routing on page 2309
Optimized VXLAN Replication Using Underlay Multicast on page 2313
Unicast and Multicast VXLAN on page 2317
Virtual Network Flood Modes on page 2318
Assisted Replication on page 2319
OSPFv2 VXLAN Extensions on page 2324
Multiprotocol Border Gateway Protocol (MBGP) Support for VXLAN on page 2325
Address Resolution Protocol (ARP) Learning over Tunnels on page 2326
Address Resolution Protocol (ARP) Suppression on page 2326
Dynamic Virtual Networks on page 2327
Edge Automation on page 2328
VXLAN Learning on page 2329
Load Balancing on page 2330
Quality of Service on page 2330
Redundant Access on page 2331
Statistics on page 2332
VXLAN + VMAN Customer Edge Ports Support on page 2333
Policy and VXLAN on page 2333
Configuring VXLAN on page 2333
Configuration Samples on page 2337

VXLAN is a network virtualization technology that leverages existing Layer 3 infrastructures to create tenant overlay networks. VXLAN addresses the requirements of a multi-tenant data center network infrastructure by:

- Increasing virtual network scalability to 16 million instances. This allows for tenant *VLAN (Virtual LAN)* isolation whereby multiple tenants can manage their own VLAN/VMAN and MAC address spaces.
- Adding an encapsulation that effectively hides VM MAC addresses from the physical network that results in physical networking devices to have smaller MAC and IP table sizes.
- Allowing for Layer 2 adjacency across IP networks by which DC network operators protect their investment in their current infrastructure. Additionally operators can distribute traffic loads across links efficiently using Layer 3 ECMP.

VXLAN Overview

VXLAN is a Layer 2 overlay scheme over a Layer 3 network. Overlays are called VXLAN segments and only VMs and physical machines (tenents) within the same segment have Layer 2 connectivity. VXLAN segments are uniquely identified using an identifier called the VXLAN Network Identifier (VNI). The VNI is a 24-bit identifier; therefore, an administrative domain can support up to 16 million overlay networks.

As the scope of the MACs originated by tenants is restricted by the VNI, overlapping MAC addresses across segments can be supported without traffic leaking between tenant segments. When a tenant frame traverses a VXLAN overlay network, it is encapsulated by a VXLAN header that contains the VNI. This frame is further encapsulated in a UDP header and L2/L3 headers.

VXLAN can add up to a 54-byte header to the tenant VM's frame. For VXLAN to work correctly, this requires that the IP MTU be set to at least 1554 bytes on the network-side interfaces. IP MTU of 1554 should also be set on all transit nodes which carry VXLAN traffic. The point at which a tenant frame is encapsulated (or decapsulated) is referred to as a VXLAN Tunnel Endpoint (or VTEP). VTEPs are typically located on hypervisors but may also be located on physical network switches. Network switches that act as a VTEP are referred to as VXLAN gateways.

The role to encapsulate/decapsulate a frame is performed by a VXLAN Tunnel Endpoint (VTEP), also referred to as a VXLAN gateway. A VXLAN gateway can be a Layer 2 gateway or Layer 3 gateway depending on its capacity. A Layer 2 gateway acts as a bridge connecting VXLAN segments to *VLAN* segments. A Layer 3 gateway performs all that of Layer 2 gateway, and capable of routing traffic between tenant VLANs/VMANs.

At tunnel initiation, a gateway looks up the destination MAC address of the frame received from the tenant VM. If the MAC address to remote VTEP IP binding is known, the gateway adds the VXLAN header and the IP/UDP header to the frame and forwards toward the DC network. A gateway node that terminates a tunnel removes the encapsulation headers from the packet and determines the bridge domain of the inner frame by examining the VXLAN network identifier (VNID) received in the VXLAN header. The gateway then looks up the inner MAC destination address (DA) in the tenant VLAN's/VMAN's filtering database and decides either to flood or forward the frame to tenant ports.

The VXLAN segments with the same virtual network ID form a virtual network with one Ethernet broadcast domain.

Note
Universal switches support Layer 3 gateways.

Supported Platforms

VXLAN is supported on Universal switches, and stacks with Universal slots only.

Note
4120, 4220, and 5120 Series do not support VXLAN.

Limitations

The following capabilities are not supported in ExtremeXOS:

- Ability to assign more than one VNI to a virtual network.
- IPv6 addresses for local and remote VTEPs.
- Assigning source IP for VXLAN gateway encapsulation:
  - Per
  - Per virtual network or VNI
- Support for adding more than one tenant VLAN/VMAN per VNI.
- A physical port being part of both a tenant VLAN/VMAN and an underlay (Network) VLAN.
- Support for heterogeneous stack environments where at least one of the stack nodes is not VXLAN capable.
- More than one next-hop on a (network) port.
- Multicast underlay IP network, except PIM-SSM.
- Multiple VRs.
- With VXLAN for 5320 and 5420 series switches, MACSec-enabled shared ports cannot be used as a VXLAN tenant/network port.

Interactions with Existing Features

Feature/Capability	Tenant Network	Underlay Network	Rest of the Switch
VLAN with: Multiple C-Tags on the same port or different C-Tags on different ports	Future	Future	No new restrictions
Port-Specific Tag (PSTag)	ExtremeXOS 31.2 or later	Not supported	No new restrictions
MAC-based and Protocol-based VLANs	Not supported	Not supported	No new restrictions
VMANs	Supported	Not supported	No new restrictions
CEP	Not supported	Not supported	No new restrictions
Configuring *LAG (Link Aggregation Group)* on ports (static and LACP)	ExtremeXOS 21.1 or later	ExtremeXOS 21.1 or later	No new restrictions
Configuring on ports	ExtremeXOS 21.1 or later	ExtremeXOS 21.1 or later	ExtremeXOS 21.1 or later
Limit learning and MAC locking	ExtremeXOS 31.2 or later	Not supported	ExtremeXOS 21.1 or later
Configuring IP (v4/v6) address on a VLAN	ExtremeXOS 21.1 or later	ExtremeXOS 21.1 or later	No new restrictions
Enabling IP and IPMC forwarding	ExtremeXOS 22.5 or later	ExtremeXOS 21.1 or later	No new restrictions

Feature/Capability	Tenant Network	Underlay Network	Rest of the Switch
Interface virtual router configuration	ExtremeXOS 22.5 or later	ExtremeXOS 21.1 or later	No new restrictions
Spanning tree (802.1d, RSTP, *MSTP (Multiple Spanning Tree Protocol)*, EMISTP, and PVST+)	Not supported	ExtremeXOS 21.1 or later	No new restrictions
Ring protocols (*EAPS (Extreme Automatic Protection Switching)* and ERPS)	ExtremeXOS 21.1 or later	ExtremeXOS 21.1 or later	No new restrictions
VRRP (Virtual Router Redundancy Protocol)	ExtremeXOS 22.5 or later	ExtremeXOS 21.1 or later	No new restrictions
ESRP	Not supported	ExtremeXOS 21.1 or later	No new restrictions
IGMP (Internet Group Management Protocol) Snooping	Not supported.	ExtremeXOS 21.1 or later	No new restrictions
Unicast routing protocols	Not supported.	ExtremeXOS 21.1 or later	No new restrictions
PIM (SM, DM, SSM)	Not supported	ExtremeXOS 21.1 or later	No new restrictions
PIM Bidir	Not supported	Future	No new restrictions
MVR	Not supported	Future	No new restrictions
MPLS (Multiprotocol Label Switching)	Not supported	Not supported	No new restrictions
VPLS Service VLAN configuration	Not supported	Not supported	No new restrictions
DCBX	Not supported	Not supported	No new restrictions
ETS	ExtremeXOS 21.1 or later	ExtremeXOS 21.1 or later	No new restrictions
Extreme Network Virtualization (XNV)	Not supported	Not supported	Restricted
Private VLANs, VLAN aggregation, VLAN translation	Not supported	Not supported	No new restrictions
Identity Management	Not supported	Not supported	Restricted
IP security (*DHCP (Dynamic Host Configuration Protocol)* Snooping, ARP lockdown)	Not supported	Not supported	No new restrictions
802.1ag CFM and Y.1731 performance monitoring	Not supported	No supported	No new restrictions

Feature/Capability	Tenant Network	Underlay Network	Rest of the Switch
BFD	Not supported	Not supported	No new restrictions
AVB protocols (MVRP, gPTP, MSRP, FQTSS)	Not supported	Not supported	Restricted
Layer 2 Protocol Tunneling	Not supported	Not supported	No new restrictions
NetLogin	ExtremeXOS 32.6 or later	No new restrictions.	No new restrictions
Priority Flow Control	Not supported	No new restrictions.	No new restrictions

Note

1. MLAG requires entire switch to move to software learning mode.
2. MAC locking and limit learning features require software learning mode.
3. ExtremeXOS will require that all underlay VLANs be part of the same VR. This VR can be user-created or system-created.
4. This does not imply that VXLAN encapsulated packets will be snooped. IGMP snooping for multicast contained entirely within the operator network will work.
5. Current hardware has no support for adding an MPLS label to a VXLAN encapsulated frame. So remote VTEPs cannot be reachable through a MPLS network.
6. XNV dynamic VLANs, IDM role-based VLANs and MVRP require a global VLAN ID space and hence are not supported.
7. Single-hop BFD can be configured on operator VLANs.
8. All multicast traffic are treated as unknown.
9. Multi-chassis Link Aggregation (MLAG) does not function if the Inter-Switch Connection (ISC) port is added to an untagged tenant VMAN.
10. Non-Tenant VLANs/VMANs do not function if the port also has an untagged tenant VMAN.

Overlay Routing

Tenants may have multiple overlays across a data center network where different VLANs belonging to the same tenant are mapped to different VXLAN Network Identifiers (VNIs). Tenants require routing between the VLANs, and VXLAN gateway nodes would need to act as Layer 3 gateways that are capable of routing traffic between tenant VLANs. Inter-overlay routing involves routing:

- Routing traffic from a tenant VLAN into a tunnel with the destination overlay's VNI.
- Routing traffic from a tunnel to a tenant VLAN that is different from the tenant VLAN associated with the VNI in the received packet's VXLAN header.
- Routing traffic from a tunnel to the same or different tunnel.

A VXLAN tunnel endpoint (VTEP) is designated as the gateway for an overlay by manually configuring it or by running a first hop redundancy protocol (FHRP), such as *VRRP* on the tenant VLAN on the gateways and letting the protocol determine the placement of the router. ARP is not learned by VXLAN tunnels when the destination MAC address is the VRRP MAC address.

Note
Routing between a normal VLAN and a VXLAN tenant VLAN is supported on allExtremeSwitching Universal platforms.

Note
When MLAG and VRRP is configured on the tenant VLAN, VRRP should be active-active mode between the MLAG peers.

Routing Traffic into Tunnels

When a VTEP is configured as a Layer 3 gateway, traffic from an access VLAN port destined for an endpoint (that is reachable through a VXLAN tunnel on a different subnet) is routed by the VTEP.

In Figure 265, Layer 3 traffic is sourced from 20.20.20.1 and is destined for 10.10.10.2. The gateway for H2 to reach 10.10.10.2 is VTEP A. When traffic ingresses VTEP A, it is routed, and since the neighbor entry for 10.10.10.2 indicates that it is reachable through VTEP B, VTEP A encapsulates traffic with VNI “Red” and sends it over the tunnel to VTEP B. At VTEP B, traffic is decapsulated and is Layer 2 forwarded to H3.

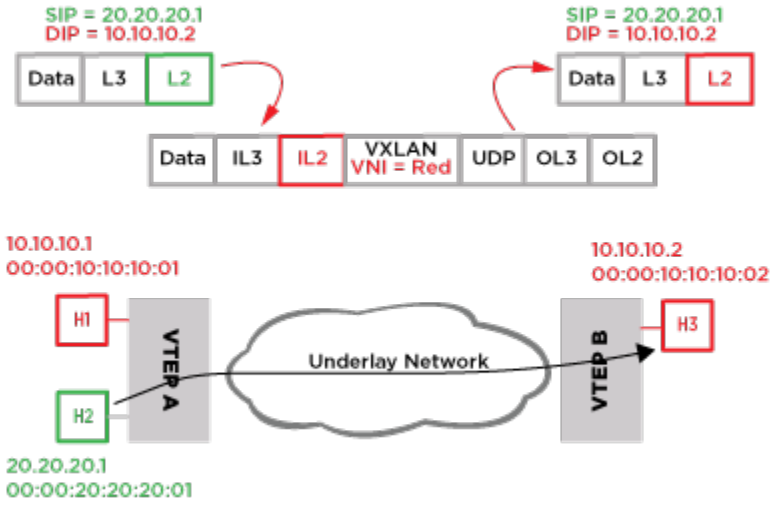


Figure 265: Routing into a Tunnel

Supported Platforms

Routing traffic into tunnels is supported on all ExtremeSwitching Universal switches.

Routing Traffic from Tunnels

When a Layer 3 gateway router is reachable across the VXLAN cloud, traffic needs to traverse the VXLAN cloud encapsulated, and the gateway at the edge of VXLAN network decapsulates traffic and would then have to route the traffic. Continuing the example from the previous subsection, the Layer 3 router for the “Red” subnet is VTEP A. Traffic in the H3 to H2 direction is VXLAN-encapsulated and Layer 2 forwarded to VTEP A. VTEP A decapsulates the traffic, performs a route lookup, and then routes traffic onto VLAN “Green” (see Figure 266).

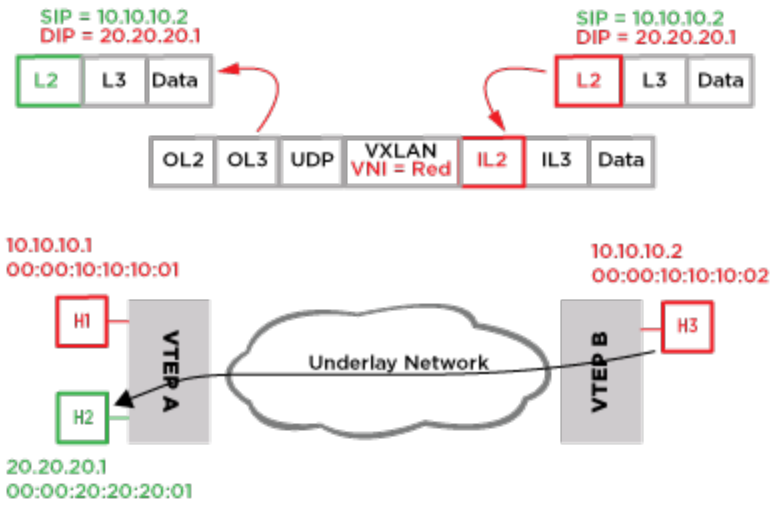


Figure 266: Routing from a Tunnel

Supported Platforms

Routing VXLAN network identifier (VNID) traffic from tunnels is supported on all ExtremeSwitching Universal switches.

Routing in and out of Tunnels (RIOT)

VTEPs acting as Layer 3 gateways would also have to support routing traffic that ingresses the VTEP on a VXLAN tunnel whose next hop is reachable through another tunnel. In Figure 267, H3 determines that the gateway to reach H4 is VTEP A. Traffic from H3 is VXLAN-encapsulated with VNI “Red” and Layer 2 forwarded to VTEP A by VTEP B. VTEP A decapsulates the traffic, performs a route lookup, and then determines that H4 is reachable through VTEP C. VTEP A now encapsulates the traffic with VNI “Green” and Layer2 forwards the traffic toward VTEP C.

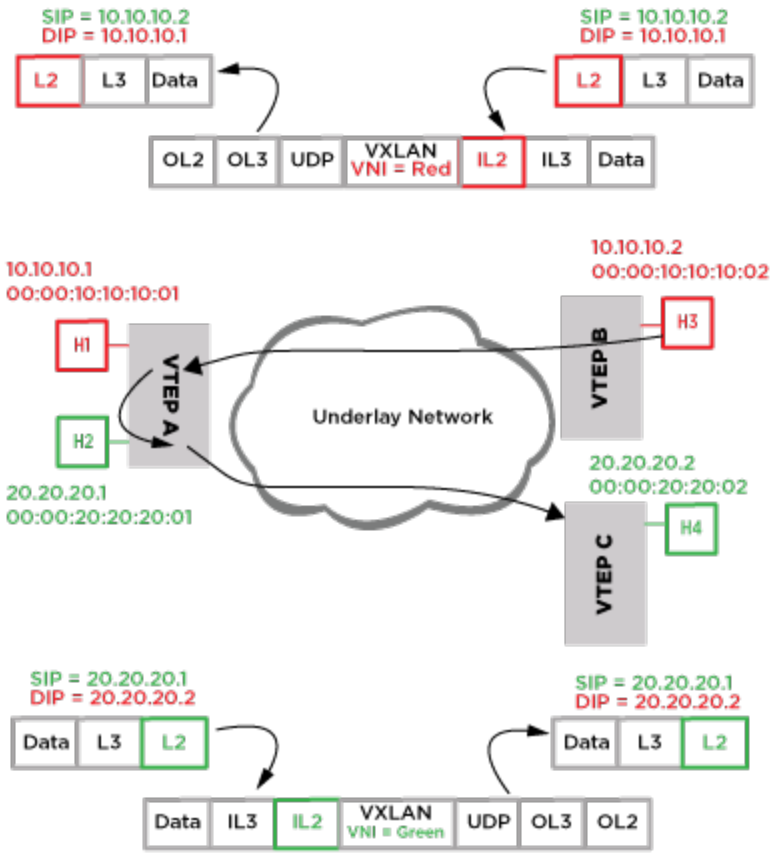


Figure 267: Routing in and out of a Tunnel

Supported Platforms

Routing traffic in and out of tunnels is supported on all Universal switches except 4120, 4220, and 5120.

Route and Neighbor Distribution

Overlay route information can be configured on gateways using:

- Static configuration (for routes that have next hops reachable over the VXLAN network).
- Overlay routing protocols such as *OSPF (Open Shortest Path First)*, *RIP (Routing Information Protocol)* etc. (not supported).

Time to Live (TTL) Processing

In ExtremeXOS, Layer 3 gateways process time to live (TTLs) information by treating a VXLAN tunnel as a single hop when an overlay IP packet traverses the tunnel.

Differentiated Services Code Point (DSCP)/IP Type of Service (ToS) processing

Underlay and overlay networks have a common Differentiated Services Code Point (DSCP)/IP Type of Service (ToS) space.

Optimized VXLAN Replication Using Underlay Multicast

Starting with ExtremeXOS 31.2, VXLAN supports underlay multicast (multicast tunnel). This feature provides support for the following:

- OSPF/BGP underlay and EVPN.
- Dynamic virtual networks.
- Bud node operation (a VTEP acting as both transit and tunnel terminating node).
- MLAG.
- Ability to configure VXLAN multicast tunnels in existing VXLAN deployments.
- Spine node can be a non-VTEP in leaf-spine topologies.
- Bandwidth preservation through optimized replication.
- Flexible choice of Multicast Distribution Tree (MDT).
- Option for automatic group address assignment for each VNI.

Head-End Replication

The default method for flooding the broadcast/unicast/multicast (BUM) traffic from VXLAN overlay to remote VTEPs is through head-end replication over VXLAN underlay. This means the originating VTEP sends a separate copy to each of the destination VTEPs over VXLAN unicast tunnel to each VTEP.

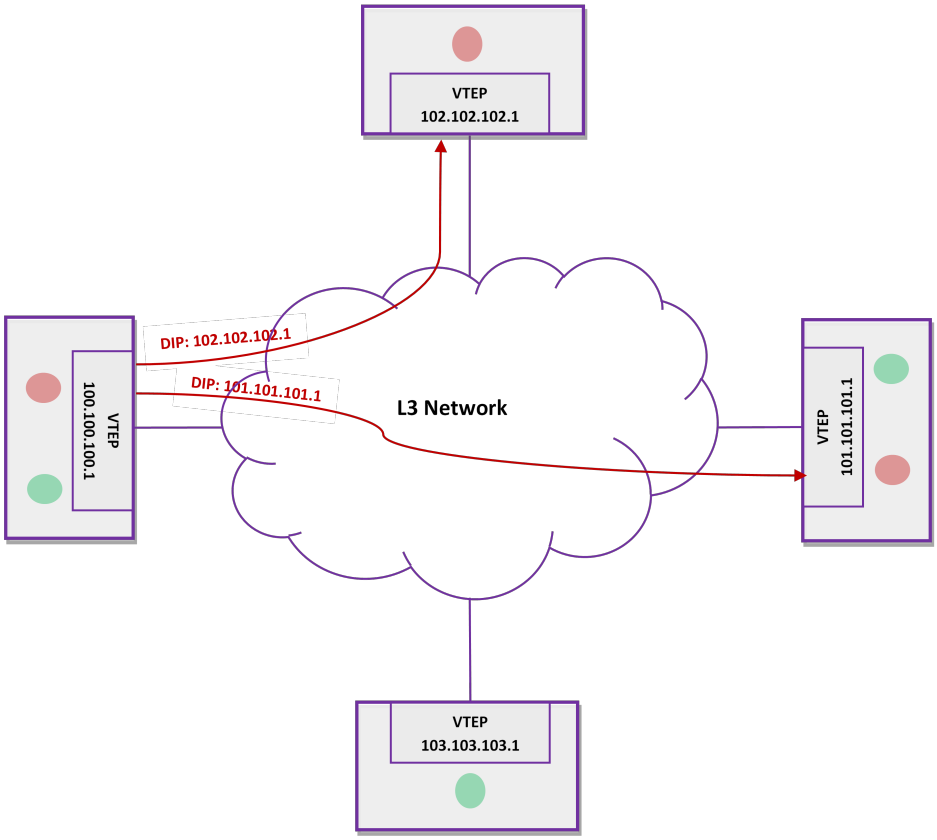


Figure 268: Head-end Replication

On receiving overlay BUM traffic, VTEP 100.100.100.1 VXLAN encapsulates and sends two copies – one copy is destined to VTEP 101.101.101.1 and the other to 102.102.102.1.

Note
RED VNI does not span to VTEP 103.103.103.1.

Underlay Multicast Distribution Tree

A multicast group address is assigned to a virtual network VNI. Each VTEP must use the same multicast group address for a given VNI. This must be administratively ensured.

When a VTEP discovers a remote VTEP, it learns the VNIs supported by the remote VTEP. The discovering VTEP triggers PIM (S, G) joins to the remote VTEP, for the multicast groups corresponding to the supported VNIs. As a result, a multicast distribution tree (MDT) is formed for each multicast group or each VNI.

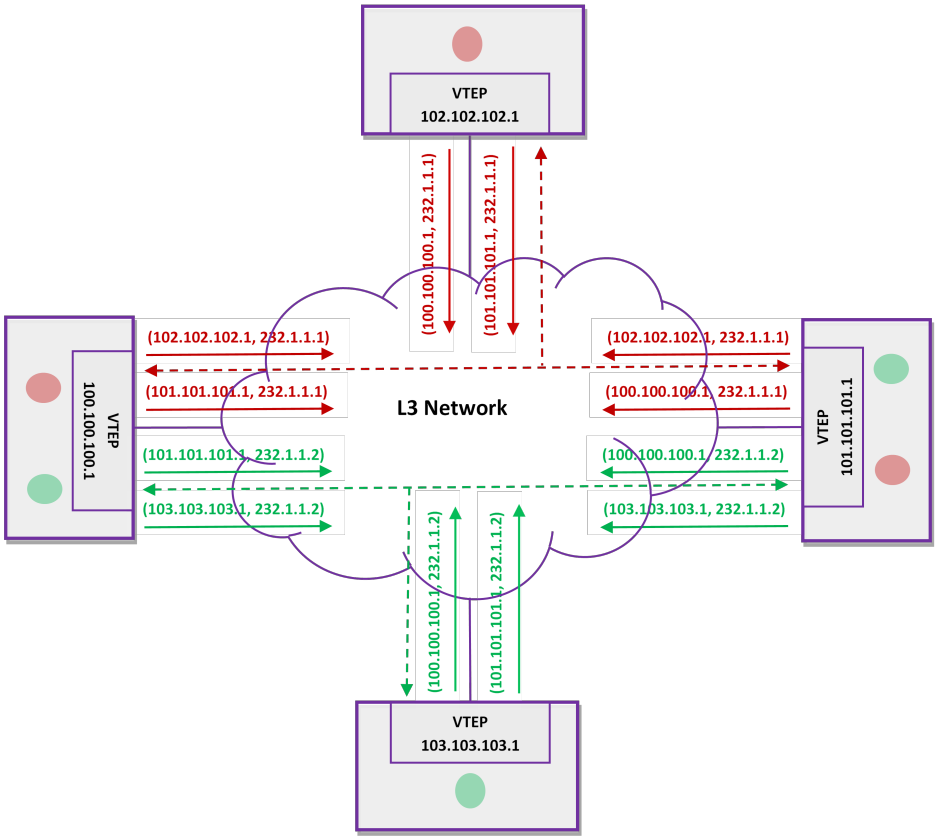
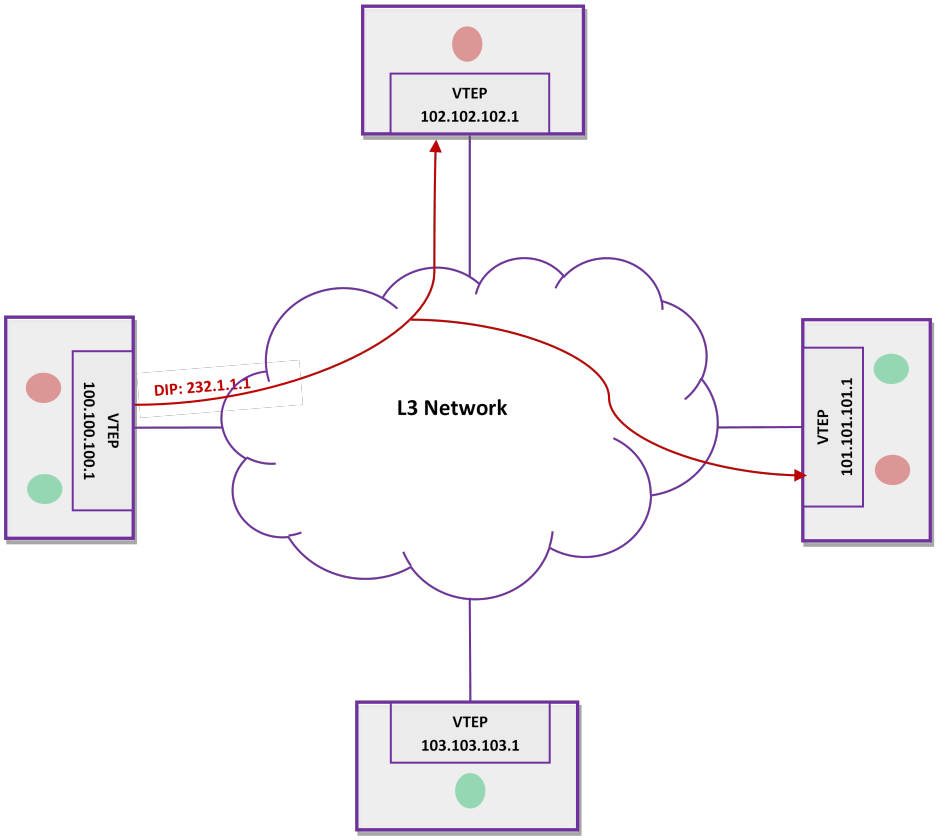


Figure 269: Underlay Multicast Distribution Tree

The RED VNI is assigned with 232.1.1.1 and GREEN VNI is assigned with 232.1.1.2. Observe the PIM joins triggered by each VTEP. The MDT for RED VNI is RED dash-tree (232.1.1.1) and the MDT for GREEN VNI is GREEN dash-tree (232.1.1.2).

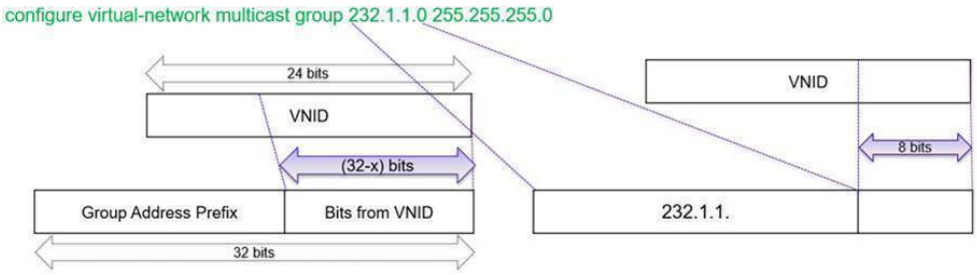
The overlay BUM traffic is VXLAN encapsulated with multicast group IP (corresponding to the VNI) as the outer destination IP and sent over the MDT. At the very best scenario, the originating VTEP sends only one copy out. The replication is done en route hop-by-hop by the routers in the multicast tree present in L3 network. The routers in the L3 network must support PIM-SSM but need not be VXLAN aware.

The following figure illustrates how VTEP 100.100.100.1 VXLAN encapsulates the traffic with destination IP as 232.1.1.1 and sends one copy out. The traffic flows across the L3 network over the MDT and reaches 101.101.101.1 and 102.102.102.1:



Automatic Group Assignment and MDT Choices

It is possible to automatically assign multicast group address for each VNI. The group address is picked from a user configured address range. The address is derived as depicted in the below figure. The group prefix is obtained by AND'ing group address and mask. The group suffix bits come from the VXLAN network identifier (VNID).



Using the command featured in the previous figure, it is possible to choose different variants of MDT as shown here:

1. Each virtual network VNI can be assigned with discrete multicast group address. Meaning, each virtual network uses a dedicated MDT. The following command auto-assigns separate group address to each VNI:

   ```
   configure virtual-network multicast group 232.0.0.0 255.0.0.0
   ```

2. Single multicast group address can be used for all virtual network VNIs. A single MDT is used for overlay BUM traffic on all VNIs. The following command auto-assigns 232.1.1.1 to all VNIs:

   configure virtual-network multicast group 232.1.1.1 255.255.255.255

3. A combination of (1) and (2) can be used. That is, a cluster of VNIs can use one multicast group address. Another cluster can use a different multicast group address. The following command forms four cluster of VNIs and uses four shared MDTs:

   configure virtual-network multicast group 232.1.1.0 255.255.255.252

Virtual Network VNIs	Shared MDT
1, 5, 9, 13, 17, and so on	232.1.1.1
2, 6, 10, 14, 18, and so on	232.1.1.2
3, 7, 11, 15, 19, and so on	232.1.1.3
4, 8, 12, 16, 20, and so on	232.1.1.0

In any of the previous options, a single VNI cannot use more than one multicast group address.

Supported Platforms

All ExtremeSwitching Universal platforms.

Limitations

This feature has the following limitations:

- Supported only for overlay BUM traffic.
- Supported only with PIM-SSM.
- BUD node operation is not supported with MLAG.
- Multicast group range used for this feature must not be configured/used for regular multicast traffic.

Note
This feature should not be used with Assisted Replication.

Unicast and Multicast VXLAN

In multicast VXLAN, the VNI is mapped to a multicast group and multicast tunnels are used to distribute broadcast, unknown unicast and multicast (BUM) tenant traffic to remote endpoints (VTEPs). This requires that the Layer 3 network supports multicast.

Unicast VXLAN uses unicast tunnels, and the BUM traffic is head-end replicated to each of the remote endpoints

Virtual Network Flood Modes

In ExtremeXOS, a virtual network operates in one of the following flood modes:

- Flood mode standard (default)
- Flood mode explicit
- Flood mode multicast

These modes determine the way the remote endpoints are created/learned, and the way the tenant BUM traffic is handled.

Flood Mode Standard

- The remote endpoints dynamically are learned using *OSPF* VXLAN extensions (see OSPFv2 VXLAN Extensions on page 2324) or Multiprotocol BGP (MBGP) support for VXLAN (see Multiprotocol Border Gateway Protocol (MBGP) Support for VXLAN on page 2325). Alternately, remote endpoints can be statically configured and attached to a virtual network using the CLI.
- The tenant BUM traffic is head-end replicated to each of the configured/learned remote endpoints on that virtual network.

For a configuration sample, see Configuration Example for Flood Mode Standard on page 2337.

Flood Mode Explicit

- The remote endpoints can only be statically configured by attaching them to BUM *FDB (forwarding database)* entries using the CLI.
- The tenant BUM traffic is head-end replicated to each of the configured remote endpoints for the corresponding FDB entry.
- By explicitly adding BUM FDB entries, this mode provides flexibility in splitting the BUM entries to individual broadcast, unknown unicast, and unknown multicast entries.

For a configuration sample, see Configuration Example for Flood Mode Explicit on page 2337.

Flood Mode Multicast

- The remote endpoints are learned in the same manner as in flood mode standard.
- The tenant BUM traffic is replicated by the underlay multicast distribution tree.

Note
This mode requires L3 multicast support on the underlay network.

Note
The only difference from standard flood mode is the way the tenant BUM traffic is handled.

Note
This mode should not be used with Assisted Replication.

Assisted Replication

The following sections explain the Assisted Replication feature.

Assisted Replication Overview

Assisted Replication works in conjunction with the BGP control plane and the "ipv4-vxlan" proprietary address-family. When this address-family is enabled the configured replication role and local tunnel endpoint are advertised using BGP updates, and each router selects the active replicator such that each router selects the same device. If a device is configured as a replicator, but is not selected it assumes the role of a leaf.

Optimized Ingress Replication for EVPN is an IETF RFC (currently in draft 6, Standards Track) that enhances the ingress replication model of broadcast, unknown, unicast, and multicast (BUM) traffic handling without putting any further burden on the underlay. For example, using multicast-based trees for BUM traffic required a multicast protocol in the underlay, such as PIM. With Optimized Replication all BC/MC frames are still unicasted through the underlay, requiring no new protocols for packet delivery, but done so in a more efficient manner. Assisted replication is an enhancement to BUM traffic handling within the ExtremeXOS *VXLAN (Virtual Extensible LAN)* gateway feature.

Note
ExtremeXOS VTEPs should *not* be configured as leaf.

Note
This feature should not be used with Optimized VXLAN Replication using Underlay Multicast.

To get a better understanding how the assisted replication works, you must first understand what head-end replication is.

Head-End Replication

ExtremeXOS handles BUM traffic in VXLAN environments using head-end replication. Head-end replication means that each local tunnel end-point (LTEP) must replicate the frame and VXLAN encapsulate it for each remote tunnel end-point (RTEP) on the virtual network (VNI). The primary benefit of head-end replication is that it imposes no new restrictions or requirements on the underlay. All frames are unicast IP VXLAN frames, just like the normal/learned VXLAN frames. In Figure 270, you can see for a given switch what the head-end replication looks for this VNI. In this case, the total copies sent by the ingress switch is N+M+3, all of which must traverse the uplinks from the ingress switch.

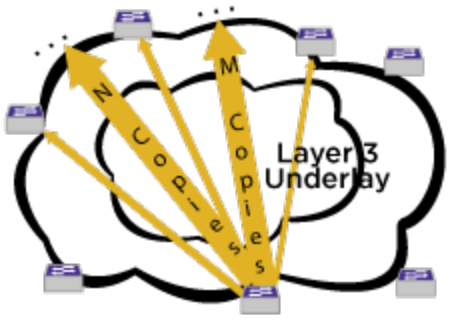


Figure 270: Example Head-End Replication Topology

Assisted Replication

Assisted replication can be thought of as moving the head-end replication from every end-point to a small set of replicator nodes. The end result is the same, however, as each end-point receives a unique copy of the BC/MC frame (unique because it is unicasted directly to the end-point). Assuming the replicators are placed in the spine or core of the network, the actual number of frames traversing any single link can be greatly reduced.

In Figure 271, the ingress switch needs to only send one copy of the BC/MC frame to the replicator. The replicator is then responsible for replicating the frame to all the end-nodes in the VNI. One switch can be the replicator for all VNIs in the VXLAN network, or multiple switches can be configured as the replicator for a specific set of respective VNIs.

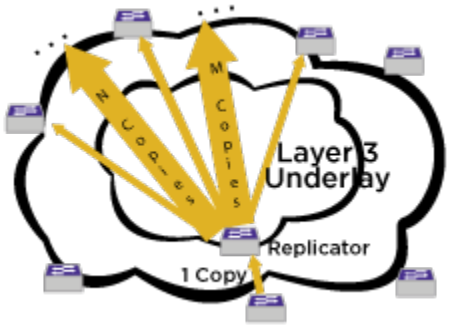


Figure 271: Example Assisted Replication Topology

There is no restriction on the number of replicators used in a topology. Each VNI can have its own dedicated replicator, or multiple VNI can share a replicator as shown in Figure 272. This would be a network operator decision based on the amount of BC/MC traffic generated on a particular VNI. Note that both Figure 271 and Figure 272 show the path from one particular end-point to the other end-points, when in fact these paths are bi-directional and all end-points on a VNI send BC/MC to (and receive BC/MC

from) the replicator. However, when the replicator role per VNI is split up, the restriction applies that a particular switch can only operate in one role: replicator or leaf.

Note
In non-EVPN environments there is no control plane learning, and there is no way for endpoints to learn the location of the user MAC addresses. In EVPN environments, which have a control plane learning model, leaf nodes learn remote addresses from the EVPN control plane, and learning is disabled on the tunnels.

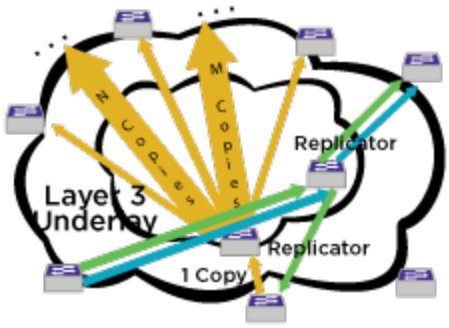


Figure 272: Assisted Replication with Multiple Replicators

Assisted Replication Roles

Each switch participating in Assisted Replication is given a role. The role defines whether the switch acts as a regular node (meaning Assisted Replication is not used), a replicator node, or a leaf node:

- **Regular network virtualization edge node (RNVE)**—RNVE is a regular network virtualization edge node, meaning it does not participate in the Assisted Replication, but does participate in VXLAN. This is the default role for a switch. If not defined as leaf or replicator, the switch performs normal head-end replication. RNVE nodes cannot be mixed with Assisted Replication (leaf or replicator) nodes.
- **Replicator node**—The replicator is the switch that performs the BC/MC replication for configured VNIs. There can be different replicators defined for any set of VNIs. The replicator is responsible for delivering BC/MC frames to all end-points (leafs) for a given VNI. There can only be one replicator configured per VNI.
- **Leaf node**—The leaf nodes are all other nodes participating in Assisted Replication in a VNI, excluding the replicator. Leaf nodes forward BC/MC frames to the defined replicator for the virtual network, and receive BC/MC frames from the replicator (originated from other leafs). If configured as a leaf node, the switch is configured with the IP address of the replicator for the VNI. The role of leaf is not implied on a switch. You must configure it for this role, since the default role is RNVE (meaning no Assisted Replication).

For information about configuring Assisted Replication, see Configuring Assisted Replication on page 2336.

Assisted Replication Benefits

There are a couple benefits of deploying the Assisted Replication feature:

- **Reduces capacity edge nodes/higher capacity replicators**—Using Assisted Replication allows for the edge nodes to be designed with lesser VXLAN replication capabilities, since they may not need perform broadcast or multicast replication. At the same time, switches that have exceptional replication capabilities, can be targeted for use in the role of replicator.
- **Leaf-to-spine (or edge-to-core) bandwidth utilization**—In a traditional VXLAN environment with head-end replication, the leaf/edge node has to replicate a frame to multiple RTEPs over a finite number of paths/next-hops. The number of paths may be greater in a pure leaf-spine environment compared to the traditional edge/core environment, but in either case, the bandwidth consideration from leaf-to-spine or edge-to-core is typically scaled based on requirements of nodes that will be attached. The more BC/MC frames that need be replicated from the node, the more of this precious bandwidth it consumes as the number of RTEPs increases. At the same time, as more and more replication transmits are sent out a given port, the latency of the latter frames increases. Using Assisted Replication, if the replicator is placed in a strategic location, the bandwidth utilization is reduced. For example in a leaf-spine topology where a spine has a link to every leaf, then the replication is no more than one frame out any link. In a non leaf-spine topology, the replicator could be located on a device that has plenty of bandwidth and links to the edges of the network.

For information about configuring Assisted Replication, see Configuring Assisted Replication on page 2336.

Assisted Replication Operation

The Assisted Replication feature operates as follows.

Configuring Roles

The first step for using the Assisted Replication feature is to configure the switches as replicator. After this step is complete, the Assisted Replication feature is ready to operate.

Node Discovery

Assisted Replication depends on the discovery of tunnel end-points. There are currently a few different methods for this to happen within ExtremeXOS, including static configuration, or EVPN. The Assisted Replication feature is part of the OTM process and learns of the remote tunnel end-points (RTEPs) directly from there, and does not care how the end-points are discovered.

Frame Encapsulation

Assisted Replication feature only uses VXLAN Unicast-IPv4 encapsulation.

Leaf Operation

The leaf nodes have three general task:

- Forward frames received on a local access port to the other local access ports that are a member of the underlying tenant VLAN.
- Forward, by VXLAN encapsulation, the frame to the selected replicator following the VLAN-to-VNI mapping.
- Receive a VXLAN encapsulated frame from the replicator and de-cap it, and then forward it to the local access ports that are a member of the underlying tenant VLAN.

When a node is originally configured for VXLAN it will build BUM flood lists that include all local access ports for the tenant VLAN as well as all remote TEPs participating in the VNI. When the node is then configured as a leaf, the system modifies the underlying BC/MC flood list to still include the local access ports for the tenant VLAN, but then only add the selected replicator as a tunnel destination.

Note
If a node is configured as a leaf, but it has no selected replicator (either it was not configured, or no node has been configured to advertise as such), no replication takes place, except to local access ports. This occurs to prevent duplication of frames that can occur when combining head-end with Assisted Replication.

For information about configuring Assisted Replication, see Configuring Assisted Replication on page 2336.

Assisted Replication with VXLAN Access Points

Access points (APs) have LTEPs defined on each AP, which connect to one, and only one, replicator. Unicast and BUM traffic traverses the same path up the *VXLAN* tunnel to the replicator. Unicast traffic is forwarded out the correct path towards the destination similar to the way RIOT works (see Routing in and out of Tunnels (RIOT) on page 2311), but in this case the VXLAN traffic is not routed in/out, but rather switched in/out. BUM traffic is replicated to all other APs on the VNI, as well as local access ports. The restriction of a single replicator allows the replicator to have local access ports without negative effect.

Similar to regular Assisted Replication functionality, the role of the replicator in an AP topology includes replicating BUM traffic on behalf of the sending VXLAN VTEP (AP in this case). The APs do not have the necessary hardware or CPU cycles to perform the VXLAN BUM replication, so the replicator offloads the work load from the AP. At the same time, the replicator is also the only RTEP known by the APs, so all unicast VXLAN traffic is also forwarded through the replicator to its ultimate RTEP destination.

Assisted Replication Supported Platforms

Assisted Replication is supported on all Universal switches:

Assisted Replication Limitations

In addition to restrictions imposed by VXLAN (or EVPN), the following are currently *not* supported:

- Mixed replicator/leaf functionality. A switch can be configured as a replicator *or* a leaf for all virtual network identifiers (VNI).
- Assisted replication control plane (as defined in *draft-ietf-bess-evpn-optimized-ir*).
- Local attachment (access interface) for VNI on the replicators when there is more than one active replicator per VNI.
- Mixing assisted replication with head-end replication per VNI.
- Assisted replication when not used in an EVPN or VXLAN topology.
- Combining assisted replication with IGMP/MLD snooping on the tenant VLAN, or PIM on the tenant VLAN.

Additional limitations:

- Without the control plane, a replicator must assume all other end-points are leafs.
- Stacks that do not support VXLAN are not supported.
- Only a single replicator per VNI is supported.
- The replicator can support a maximum of 250 attached leaf nodes.

OSPFv2 VXLAN Extensions

ExtremeXOS leverages *OSPF* to advertise and learn VTEPs dynamically in a VXLAN network. OSPFv2 advertises a triplet of VNI/Endpoint IP Address/Advertising Router ID through OSPFv2 domain using type 11 opaque link state advertisements (LSAs). The OSPFv2 VXLAN LSA link state ID uses opaque type 128. The remaining 24 bits of the field are set to the VXLAN VNI. Each locally configured VNI corresponds to a single opaque LSA advertised by the router. The OSPFv2 VXLAN LSA payload contains one top level TLV which specifies the locally configured IPv4 endpoint address on the advertising router.

The OSPFv2 VXLAN opaque LSA is only advertised if ospf vxlan-extensions is enabled using the `enable` **`ospf vxlan-extensions`** command.

Note

- OSPF vxlan-extensions can only be enabled when OSPFv2 is disabled.
- The remote endpoints learned by OSPF are not saved to the configuration.
- The OSPFv2 VXLAN opaque LSA is only advertised if OSPF VXLAN extensions is enabled.

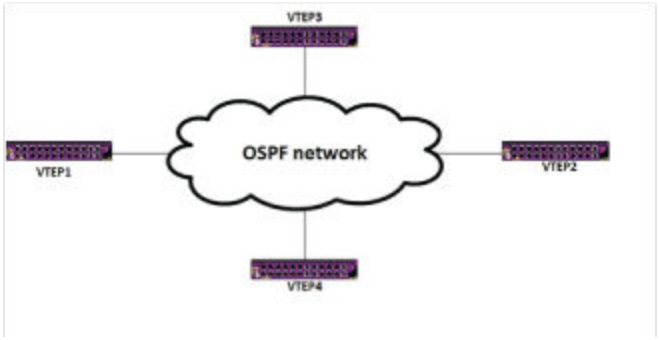


Figure 273: OSPFv2 VLXLAN Extensions

Note
Routers in OSPF network should be capable of transmitting type 11 opaque LSAs defined in *RFC5250*.

Multiprotocol Border Gateway Protocol (MBGP) Support for VXLAN

The *BGP (Border Gateway Protocol)* is a general purpose routing protocol that runs over TCP. Routes can be carried on different address family (afi) and sub-address family (safi) combinations as specified in the Multiprotocol BGP (MBGP) extensions.

The BGP implementation for ExtremeXOS supports carrying VXLAN VNI and LTEP combinations as MBGP routes using a proprietary afi/safi combination. Even though supported LTEPs are IPv4, the VXLAN routes can be carried over IPv6 peering sessions.

To enable or disable IPv4 VXLAN capability on a per peer basis, use the commands:

```
enable bgp neighbor [all |remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]

disable bgp neighbor [all | remoteaddr] capability [ipv4-unicast | ipv4-multicast | ipv6-unicast | ipv6-multicast | vpnv4 | route-refresh | ipv4-vxlan | l2vpn-evpn]
```

Limitations

- The ability to carry VXLAN routes cannot be enabled on a per peer group basis.
- BGP graceful restart is not supported on the MBGP/VXLAN feature.

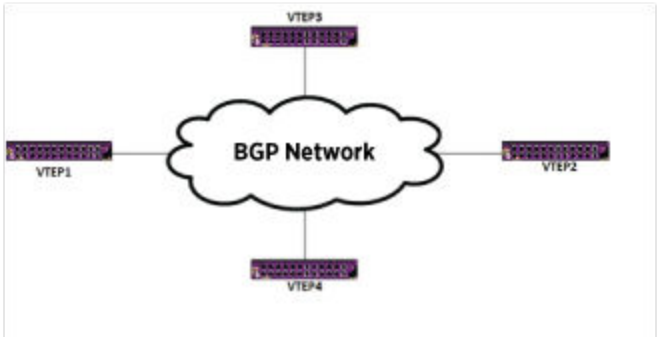


Figure 274: VXLAN over MBGP

Note
ExtremeXOS must be version 22.3 or later. Core or Premier licensing may apply (see Switch Engine v33.6.1 Licensing Guide).

Address Resolution Protocol (ARP) Learning over Tunnels

Layer 3 gateways support learning *ARP (Address Resolution Protocol)* entries for neighbors reachable through a tunnel. Gateways process ARP requests that ingress the switch from a tunnel, and respond to those requests by sending unicast ARP responses over the tunnel. Similarly, gateways broadcast ARP requests to both physical ports and tunnels and process ARP responses received from a neighbor that is reachable through a tunnel.

For VXLAN MLAG configurations, ExtremeXOS 30.5 supports checkpointing of ARP entries learned on the checkpoint-able VXLAN tenant VLAN to the MLAG peer. This helps in avoiding unwanted flooding of ARP requests when the request hashes to the MLAG peers that do not have the ARP entries learned from the tunnel. It also avoids unwanted flooding when the peer that has the ARP reboots.

Address Resolution Protocol (ARP) Suppression

ARP traffic makes up a large percentage of broadcast traffic within data centers. This traffic is even more taxing when multiple overlay networks share a common underlay network—as is the case with tunneling technologies like VXLAN. Therefore it is desirable to reduce ARP traffic.

This can be done by allowing Virtual Tunnel End Points (VTEPs) to proxy ARP requests and reply on behalf of the remote endpoint. VTEPs snoop ARP replies, exiting the virtual network tunnel to learn the remote endpoint's IP to MAC mapping. The VTEP stores this in its ARP cache for the tenant VLAN. Snooped ARP entries are viewable with the normal `iparp` commands for the tenant VLAN. VTEPS also snoop gratuitous ARPs exiting the tunnel; these may also cause the ARP cache to be updated. ARP entries are also learned from frames entering the tunnel from the tenant VLAN.

When a VTEP intercepts an ARP request it attempts to lookup the IP address in its ARP cache. If an ARP cache entry is found, the VTEP does an immediate ARP reply to the requester; this avoids flooding to both the VLAN and remote tunnel endpoints. The

source MAC in the reply is that of the remote endpoint for which the VTEP is proxying the request. On an ARP cache miss, the VTEP floods the request on the VLAN and to the remote tunnel endpoints. Additionally, the VTEP may answer an ARP request on behalf of a locally attached tenant if the tenant is on a different port than the ARP request was received on —this is proxying within the tenant VLAN.

ARP features enabled on the tenant VR (for example, gratuitous protect) may cause provoke actions for ARPs exiting the tunnel. These features should behave the same for local ARPs and ARPs processed from the tunnel. This includes configurable options such as timers and refresh.

This feature may be used even if the tenant VLAN does not have an IP interface. In that case, ARP cache entries are still learned. If an entry needs to be refreshed, the ARP request is sent with a source protocol address of all zeros. This is functionally equivalent to an ARP probe. This feature is disabled by default for configured virtual networks.

To enable/disable ARP suppression on a VXLAN tenant VLAN, use the following command:

```
configure vlan vlan_name suppress [arp-only |none]
```

To configure the filter mode for ARP suppression, use the following command:

```
configure forwarding iparp suppression filters [per-port |per-vlan]
```

To view ARP information, use the following command:

```
show iparp {ip_addr |mac | [{vlan} vlan_name | vlan vlan_list] |
permanent} {port port {vr vr_name}
```

Dynamic Virtual Networks

ExtremeXOS 30.3 introduces the ability to globally enable dynamic virtual network creation by using the following command:

```
configure virtual-network dynamic [on | off]
```

Dynamic virtual network is disabled by default. Creating or deleting BGP Auto-peering enables/disables automatic virtual network creation.

The status of this command is visible in the following command:

```
show virtual-network {vn_name | vxlan vni vni | [vlan vlan_name | vman
vman_name]}
```

When this feature is enabled, the following occurs for a new VLAN-NSI mapping:

1. A dynamic virtual network is created. The virtual network name is generated with a system reserved prefix.
2. VXLAN VNI is configured for the virtual network. The VNID is same as the value of NSI.
3. The VLAN is configured as a tenant VLAN.

When a VLAN-NSI mapping is deleted, the dynamically created virtual network along with the VNI and VLAN associations are purged.

Dynamic Virtual Networks Properties

- All restrictions for static virtual network, both configuration and functional, also apply to dynamic virtual networks.
- A dynamic virtual network is not saved to the configuration and is not persistent across reboots.
- IGMP snooping is internally disabled for a VLAN dynamically configured as tenant VLAN of a virtual network.
- When a dynamic virtual network is renamed, it becomes static and can be saved to the configuration.
- You can delete a dynamic virtual network from the CLI. It is recreated if dynamic virtual network feature is on, and the conditions are met.
- The default reserved prefix pattern for a dynamic virtual network name is SYS_VNET_xxxxxxxx, where xxxxxxxx is VNID (in the lines of dynamic VLANs – SYS_VLAN_xxxx).
- If SYS_VNET_ prefix is already in use by virtual networks in an older configuration, best effort is made to choose one of the following non-conflicting prefixes – SYS_VNET_DYN_, SYS_VNET_AUTO_, SYS_VNET_AUTOGEN_.

Edge Automation

Edge Automation connects Extreme virtual tunnel endpoints (VTEPs) with other network devices, such as switches or wireless access points (APs), by establishing a VXLAN tunnel to them, thereby receiving *VXLAN*-encapsulated traffic from the devices. Virtual networks and local tunnel endpoints (LTEPs) must already be configured on the VTEPs, and only remote virtual tunnel endpoint (RTEP) creation is dynamic.

The information about the devices (name, IP address, MAC address, etc.) is stored in a VNI-device database. The name of this VNI-device database is set using ExtremeXOS commands or ExtremeCloud Connector.

Edge Automation supports . However, no checkpointing of states between MLAG peers takes place. For MLAG to work seamlessly, VXLAN and server configuration must be provisioned identically on the MLAG peers.

Edge Automation supports stacking, and the local cache information is synced between the primary and backup. Only the primary switch is connected to the VNI-device database, and data synced between primary and backup switches is used for RTEP creation on the backup switch. After failover, the backup switch becomes the primary switch, and it becomes connected to the VNI-device database. Any notifications being processed during failover are lost.

Supported Platforms

Edge Automation works on all platforms that support VXLAN.

Limitations

- Virtual networks and LTEPs are not configured automatically. They must be configured using existing methods.
- The VNI-device database must be reachable using Mgmt-VR.
- You can only configure one database. If you need to create a new database, the existing database must be deleted first before adding the new database.
- This feature works only with Extreme Campus Controller. It does not work with any third-party applications.
- Any existing limitations on dynamic RTEPs apply to RTEPs created by Edge Automation.
- If a database's password is changed, and if the same is configured on ExtremeXOS, then the connection between ExtremeXOS and the database is re-established, causing a brief loss of connectivity.
- If a network device supports multiple IP addresses, the first IPv4 address is used for RTEP creation.
- Traffic between ExtremeXOS and the database is un-encrypted.

For information about configuring Edge Automation, see Configuring Edge Automation on page 2334.

VXLAN Learning

Filtering databases on the gateways need to be populated (either through configuration or by dynamic learning) with remote MAC-to-VTEP bindings on the gateways to avoid unnecessary replication of tenant traffic.

Dynamic Learning

Dynamic learning of remote MACs involves:

- VTEP discovery—When a VNI is configured on a gateway, other VTEPs in the network would somehow have to discover the existence of the VTEP(s) this gateway is configured with.

  Note
  VTEP discovery is only supported using *OSPF* extension. But not through receiving an encapsulated frame.

- Remote MAC to IP—When an encapsulated frame is received from a remote VM, a VTEP needs adds an entry in its filtering database that binds the remote VM's MAC to the IP address of the remote VTEP

VXLAN supports standard and explicit-remotes flooding modes:

- Explicit-remotes mode—VTEPs participating in a VNI are created statically either by the user or the controller.
- Standard mode—OSPF and BGP are used to signal existence of other VTEPs. Traffic from unknown VTEPs is dropped and MAC to IP bindings are never learned.

Additionally the MAC-to-remote IP binding is added to the filtering database associated with the VNI's overlay network. The received encapsulated frame can either be addressed to the VTEP directly (in the case of known unicast) or to a multicast group that the VTEP joined.

Gateways will need to age remote VTEPs as done with MAC-to-IP bindings to ensure that the number of remote VTEPs does not remain monotonically non-decreasing over time.

A VM's MAC-to-IP binding may change if it has been migrated using Vmotion. As with *FDB* learning, VTEPs need to dynamically detect these moves based on received traffic and update their filtering databases.

Load Balancing

VXLAN allows networks to leverage Layer 3 ECMP to efficiently distribute traffic loads across the network. To enable a level of entropy for load balancing on VM to VM traffic, ExtremeXOS derives the source UDP port in the encapsulated frame from the tenant packets. Transit nodes need to be configured to use appropriate mechanism to exploit the entropy.

```
configure sharing address-based custom ipv4 L3-and-L4
configure forwarding sharing L3_L4
```

Note
The Inner packet tag (tenant tag) is not used in entropy calculation.

Quality of Service

VXLAN gateway uses a DSCP value of 0 when the tenant packet is not IP. Otherwise, it copies the tenant DSCP value to the outer IP header. In order to overwrite this, user must configure an *ACL (Access Control List)* rule. This ACL is general and applied to all virtual networks. The following is an example of this ACL rule:

```
entry map {
      if {
          protocol udp;
          destination-port 4789;
      }
      then {
          qosprofile qp3;
          replace-dscp;
          replace-dot1p;
 }
 }
```

Then, the following command is used to put the outer DSCP value to a fixed value of 32:

```
configure diffserv replacement qp3 code-point 32
```

As indicated in the example above, the dot1p value can also be replaced. In the absence of the ACL, VXLAN gateway copies the tenant dot1p value to the network. The replacement happens on the outer headers as the example above illustrates. *QoS (Quality of Service)* examination (dot1p or diffserv) happens on the first header. On the tenant port, it is examining the ethernet and IP header of tenant frames. On the

network port, it is examining the outer Ethernet and IP header. There is no change to the existing support for treatment of packets, such as rate-limiting and queuing.

Redundant Access

Because the VXLAN gateway node can become a single point of failure, it is important to dual-home devices or clusters to a pair of gateways. While there may be multiple mechanisms to accomplish this, using is the best solution due to its simplicity and because it imposes no additional constraints on the tenant devices except that they support *LAG*.

Multi-switch Link Aggregation (MLAG)

The following topology has a pair of server clusters that are dual-homed to a pair of VXLAN gateways with three tenants:

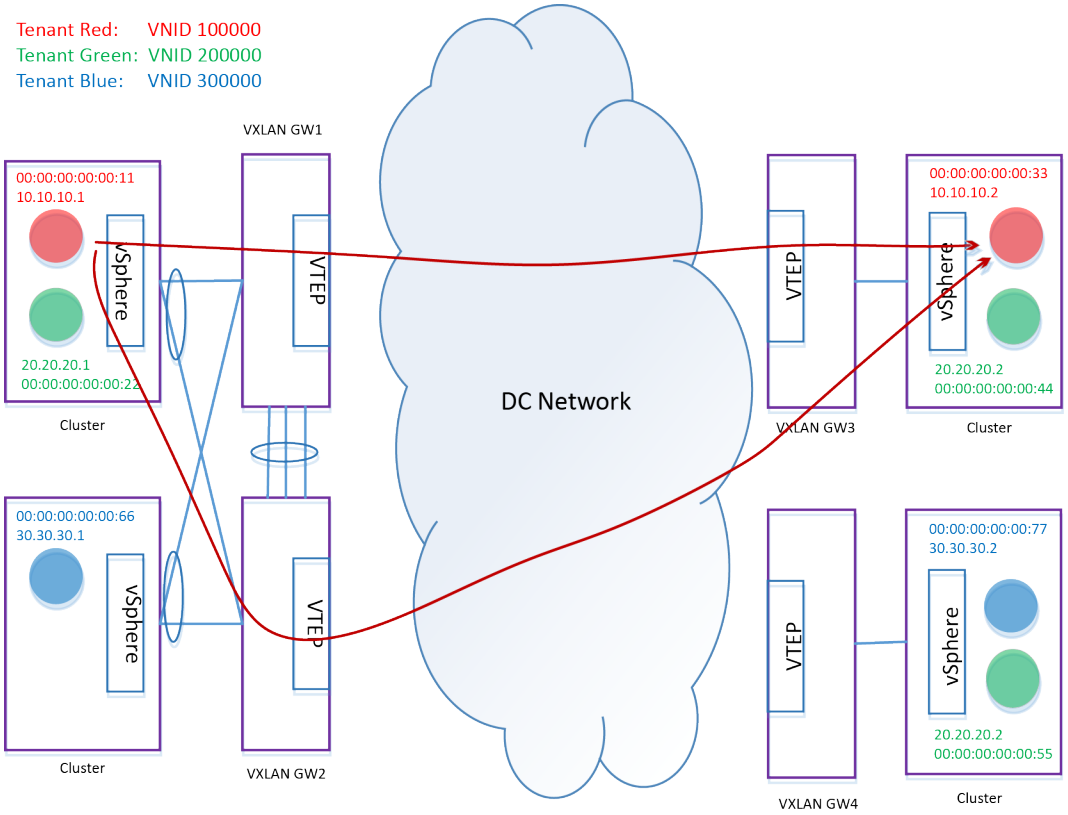


Figure 275: VXLAN Gateways with Three Tenants

Traffic from the tenant VMs that are part of an can hash toward either gateway node. This can result in GW3 learning the same VM's MAC from different VTEPs. To avoid MAC toggling on remote VTEPs, ExtremeXOS requires that users configure a special local tunnel endpoint address on MLAG nodes that will be used for tenant *VLAN*s/VMANs that have MLAG ports.

For a configuration example, see Configuration Example for MLAG on page 2338.

Additional considerations:

- Since the tenant VLAN/VMAN IDs can be reused, ExtremeXOS needs to allow multiple VLANs/VMANs with the same tag to be dual homed using the same MLAG peers. This requires tenant traffic be encapsulated over the ISC link to help distinguish traffic from different tenant VLANs/VMANs.
- Only one copy of BUM traffic from a tenant VM should be forwarded to the network.
- MLAG peers have a mechanism in place to block BUM traffic coming from the network to not be sent back into the network through the other peers.
- When one of the MLAG peers loses connectivity to the underlay network, reachability can be established through the ISC. This requires you to configure an underlay VLAN that includes the ISC ports and run a routing protocol over the ISC. Using the correct metric values ensures that this VLAN is not preferred by the peers to reach remote nodes during steady state and would be used only when all other connectivity to the underlay network is down.

  Note
  Multi-peer MLAG is not supported.

- MLAG with alternate IP configuration, disables the local MLAG ports when the MLAG peer is alive, but the ISC port alone goes down. This feature is used to prevent duplication of packets if MLAG peers are part of larger L2 domain. For VXLAN, MLAG peers connect to an L3 cloud on their upstream (northbound) interface and they share the same LTEP IP address. Other VTEPs use L3 ECMP to send traffic to one of the MLAG peers. With an alternate IP configured and ISC going down, traffic hashing towards an MLAG peer, which disabled its MLAG ports, results in dropped traffic. So it is recommended not to configure MLAG alternate IP feature on VXLAN VTEPs.

  Note
  In a VXLAN and MLAG environment where a node has two MLAG peers, even when MLAG alternate IP address is not configured, traffic disruption likely occurs when an ISC connection is lost between any of the MLAG peers.

- MLAG does not function if the Inter-Switch Connection (ISC) port is added to an untagged tenant VMAN.

Statistics

Per virtual network and per remote endpoint counters are supported. Both byte and packet counts are available.

Virtual network counters display the sum of all counters from all remote endpoints associated with the virtual network. Packets that arrived encapsulated on the VXLAN network port are counted on the virtual network as Rx bytes/packets (network side), and also, are counted as Tx bytes/packets (access side). For example, if 10 packets are received from remote endpoint 1.1.1.2 on virtual network vnet44, Rx Frames are 10, since 10 frames are received on the virtual network from the network interface (VXLAN encapsulated packets). Also, TX frames are 10, since 10 frames are transmitted on the virtual network tenant interface (native, non-VXLAN encapsulated packets). Similarly, on the other direction.

Remote endpoint counters display the sum of all counters from all virtual networks associated with the remote endpoint. Currently, it is not possible to display the counters for a given virtual network/remote endpoint pair.

VXLAN + VMAN Customer Edge Ports Support

The following VMAN with Customer Edge Ports (CEP) can be configured as VXLAN Tenant VMANs:

- CEP with or without VLAN Translation
- CEP with port-cvid
- VMAN ports with Egress cvid Filtering
- VMAN ports with a secondary Ethertype

Note
This support does not apply to VPEX Extended Ports.

Note
Egress filtering cannot be configured on network ports. If configured, there will be a drop in (VXLAN/MPLS) tunneled packets.

Policy and VXLAN

Version 32.6 adds support for policy and VXLAN static provisioning. Virtual networks and tenant VLANs must be statically created. NetLogin and static policy profiles facilitate user authentication through RADIUS and attach the users to the appropriate VLANs that are provisioned already.

Only a limited number of users (100 or fewer) is supported. Because the maximum number of policy profiles supported is 63, this feature can be used with a maximum of 63 virtual networks.

Limitations

- Dynamic VLANs are not supported.
- Dynamic virtual networks are not supported.
- Non-policy mode is not supported.
- RIOT is not supported.
- The policy must have TCI overwrite enabled.

Configuring VXLAN

The following section show how to configure VXLAN.

Adding and Deleting Tunnel Terminating Ports

When a port is added as *VXLAN* (virtual-network) network port, traffic from VXLAN tunnel (where it is terminated) is sent downstream to the tenants (hosts). If that port

is not part of virtual network network-port, traffic arriving from the tunnel (if tunnel ends here), traffic is not sent downstream to tenants. This port designation determines which ports can receive traffic from VXLAN tunnels. However, traffic from the tenant (host) to the virtual network (where tunnel originates) flows regardless.

To add ports that can terminate tunnels carrying VXLAN or NVGRE encapsulated traffic, use the following command:

```
configure virtual-network add network ports [all | portlist]
```

To delete ports that can terminate tunnels carrying VXLAN or NVGRE encapsulated traffic, use the following command:

```
configure virtual-network delete network ports [all | portlist]
```

Configuring Local Endpoints

The local VTEP IP address is used as source IP address in VXLAN encapsulated frames and is configured using the `configure virtual-network local-endpoint [ ipaddress ipaddress { vr vr_name } | none ]` command.

Note
A VTEP can have only one local endpoint IP.

Enabling and Disabling Remote Endpoints

To enable a remote endpoint, use the following command:

```
enable virtual-network remote-endpoint vxlan [ ipaddress ipaddress {vr
vr_name} | all ]
```

To manually disable a remote endpoint, use the following command:

```
disable virtual-network remote-endpoint vxlan [ ipaddress ipaddress {vr
vr_name} | all ]
```

To see if a remote endpoint is enabled or disabled, use the following command:

```
show virtual-network {vn_name} remote-endpoint vxlan {vni vni}
{ipaddress ipaddress {vr vr_name}}
```

Configuring Edge Automation

Creating and Deleting Databases

To create a database, use the following command:

```
create database database_name
```

To delete a database, use the following command:

```
delete database database_name
```

Configuring Databases

To add a server to a database, use the following command:

```
configure database database_name add server [host_name | ip_address]
{port port_number} {password [encrypted encrypted_password | password ]}
```

To delete a server from a database, use the following command:

```
configure database database_name delete server [host_name | ip_address |
all]
```

To change the password of a databases, use the following command:

```
configure database database_name server [host_name | ip_address]
password [encrypted encrypted_password | password ]
```

To change the time interval between retries, use the following command:

```
configure database max-retry-interval retry_interval
```

To connect or disconnect a database, use the following command:

```
configure automation edge [connect | disconnect] database database_name
```

Displaying Edge Automation Information

To view information about servers connected to databases, use the following command:

```
show database database_name
```

To view information about databases, use the following command:

```
show automation edge devices {vni vni}
```

To view information about database network devices, use the following command:

```
show automation edge devices {vni vni}
```

configure database max-retry-interval

```
configure database max-retry-interval retry_interval
```

Description

Specifies the maximum value for exponentially increasing time interval between retries for an Automation Edge remote *VXLAN* network identifier (VNI)-device database.

Syntax Description

database	Makes time interval retries remote VNI-database.
max-retry-interval	Specifies setting the maximum value for exponentially increasing time interval between retries.
retry_interval	Specifies the value for the maximum time interval between retries in seconds. The default is 600. The range is 1 to 3,600.

Default

If not specified, the maximum retry interval is 600 seconds.

Usage Guidelines

N/A.

Example

The following example sets the maximum retry interval to 800 seconds:

```
# configure database max-retry-interval 800
```

History

This command was first available in ExtremeXOS 31.1 as a demonstration feature.

This command is fully supported in ExtremeXOS 31.2.

Platform Availability

This command is available on all Universal switches supported in this document.

Configuring Assisted Replication

To configure the assisted replication role on a switch, or if the switch role is replicator, to also configure the default selected replicator for all virtual networks created subsequently, use the following command:

`configure` **`virtual-network replication-role`** `[`**`rnve`** | **`replicator`** | **`leaf`** `{`**`selected-replicator-default`** *`ipaddress`*`}]`

To view virtual network information, use the following command:

`show virtual-network {`*`vn_name`* | **`vxlan vni`** *`vni`* | `[`**`vlan`** *`vlan_name`* | **`vman`** *`vman_name`*`]}`

Configuring Policy and VXLAN

The following examples configure policy and VXLAN with various options.

```
create vlan "v123"

configure vlan v123 tag 123
```

```
create virtual-network "test" flooding standard

configure virtual-network "test" vxlan vni 1003

configure virtual-network "test" add vlan v123

configure policy profile 1 name "policy" pvid-status "enable" pvid 123
cos-status

enable netlogin ports 1 mac

configure netlogin add mac-list 00:00:01:00:00:00
```

Configuration Samples

Configuration Example for Flood Mode Standard

This configuration example shows how to configure VXLAN in flood mode standard:

Note
Only VXLAN-specific commands are shown.

```
configure virtual-network local-endpoint ipaddress 1.1.1.1
create virtual-network VNET1
configure virtual-network VNET1 vxlan vni 1
disable igmp snooping tenant1
configure virtual-network VNET1 add vlan tenant1
enable ospf vxlan-extensions
```

Remote endpoints are learned dynamically using *OSPF* VXLAN extensions. These are not saved to the configuration. Alternately, you can create a static remote endpoint using the following commands:

```
create virtual-network remote-endpoint vxlan ipaddress 2.2.2.2
configure virtual-network VNET1 add remote-endpoint vxlan ipaddress 2.2.2.2
```

Issuing the preceding commands on an existing dynamic remote endpoint changes it to a static remote endpoint. Static remote endpoints are saved to the configuration and persistent across reboots.

Configuration Example for Flood Mode Explicit

This configuration example shows how to configure VXLAN in flood mode explicit:

Note
Only VXLAN specific commands has been listed here.

```
configure virtual-network local-endpoint ipaddress 1.1.1.1
create virtual-network VNET1 flooding explicit-remotes
configure virtual-network VNET1 vxlan vni 1
disable igmp snooping tenant1
configure virtual-network VNET1 add vlan tenant1
```

The following commands dictate tenant broadcast traffic to be sent to remote endpoints 2.2.2.2 and 3.3.3.3:

```
create fdb broadcast vlan tenant1 vxlan ipaddress 2.2.2.2
configure fdb broadcast vlan tenant1 add vxlan ipaddress 3.3.3.3
```

The unknown unicast and multicast tenant traffic is controlled by using the following command options:

```
create fdb unknown-unicast ….
create fdb unknown-multicast …
```

Configuration Example for MLAG

This configuration example shows how to configure with VXLAN. In the following example:

- The MLAG switches are operating as independent *OSPF* routers. They must use different router IDs.
- The same local endpoint address is configured on both switches.
- The ISC link also functions as a routed link with a high metric. The link is also used when one switch loses all underlay connectivity.

Switch #1 Configuration

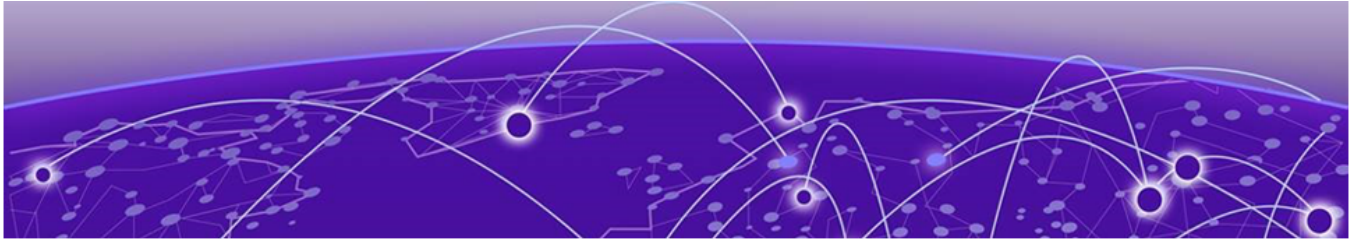
```
create vlan “ospf_loop”
enable loopback-mode vlan ospf_loop
configure vlan ospf_loop ipaddress 10.10.10.141 255.255.255.255
enable ipforwarding vlan ospf_loop
config ospf routerid 10.10.10.141
create vlan “route-isc”
configure vlan route-isc ipaddress 100.0.0.141 255.255.255.0
enable ipforwarding vlan route-isc
configure ospf vlan route-isc cost 65535
create vlan “vltep”
enable loopback-mode vlan vltep
configure vlan vltep ipaddress 10.10.10.200 255.255.255.255
enable ipforwarding vlan vltep
configure virtual-network local-endpoint ipaddress 10.10.10.200 vr “VR-Default”
```

Switch #2 Configuration

```
create vlan “ospf_loop”
enable loopback-mode vlan ospf_loop
configure vlan ospf_loop ipaddress 10.10.10.142 255.255.255.255
enable ipforwarding vlan ospf_loop
config ospf routerid 10.10.10.142
create vlan “route-isc”
configure vlan route-isc ipaddress 100.0.0.142 255.255.255.0
enable ipforwarding vlan route-isc
configure ospf vlan route-isc cost 65535
create vlan “vltep”
enable loopback-mode vlan vltep
configure vlan vltep ipaddress 10.10.10.200 255.255.255.255
enable ipforwarding vlan vltep
configure virtual-network local-endpoint ipaddress 10.10.10.200 vr “VR-Default”
```

Flood Mode Multicast

```
configure virtual-network local-endpoint ipaddress 1.1.1.1
create virtual-network VNET1 flooding multicast group 232.1.1.1
configure virtual-network VNET1 vxlan vni 1
disable igmp snooping tenant1
configure virtual-network VNET1 add vlan tenant1
enable ospf vxlan-extensions
```



Software Licensing

Extreme Networks software may contain software from third party sources that must be licensed under the specific license terms applicable to such software. Applicable copyright information is provided below.

Copyright (c) 1995-1998 by Cisco Systems, Inc.

Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies of the software and supporting documentation, the name of Cisco Systems, Inc. not be used in advertising or publicity pertaining to distribution of the program without specific prior notice be given in supporting documentation that modification, permission, and copying and distribution is by permission of Cisco Systems, Inc.

Cisco Systems, Inc. makes no representations about the suitability of this software for any purpose. THIS SOFTWARE IS PROVIDED "AS IS'" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MD5C.C - RSA Data Security, Inc., *MD5 (Message-Digest algorithm 5)* Message-Digest Algorithm

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

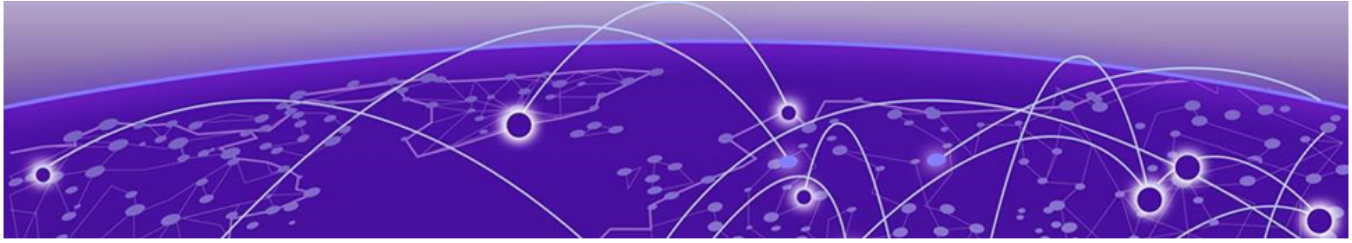
License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. *MD5* Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

$Id: md5c.c,v 1.2.4880.1 2005/06/24 01:47:07 lindak Exp $ This code is the same as the code published by RSA Inc. It has been edited for clarity and style only.



Glossary

AAA

Authentication, Authorization, and Accounting is a system in IP-based networking to control which computer resources specific users can access and to keep track of the activity of specific users over the network.

ACL

An Access Control List is a mechanism for filtering packets at the hardware level. Packets can be classified by characteristics such as the source or destination MAC, IP address, IP type, or QoS queue. Once classified, the packets can be forwarded, counted, queued, or dropped.

ad hoc mode

An 802.11 networking framework in which devices or stations communicate directly with each other, without the use of an AP.

ARP

Address Resolution Protocol is part of the TCP/IP suite used to dynamically associate a device's physical address (MAC address) with its logical address (IP address). The system broadcasts an ARP request, containing the IP address, and the device with that IP address sends back its MAC address so that traffic can be transmitted.

AS

In OSPF, an Autonomous System is a connected segment of a network topology that consists of a collection of subnetworks (with hosts attached) interconnected by a set of routes. The subnetworks and the routers are expected to be under the control of a single administration. Within an AS, routers may use one or more interior routing protocols and sometimes several sets of metrics. An AS is expected to present to other autonomous systems an appearance of a coherent interior routing plan and a consistent picture of the destinations reachable through the AS. An AS is identified by a unique 16-bit number.

ATM

Asynchronous Transmission Mode is a start/stop transmission in which each character is preceded by a start signal and followed by one or more stop signals. A variable time interval can exist between characters. ATM is the preferred technology for the transfer of images.

BGP

Border Gateway Protocol is a router protocol in the IP suite designed to exchange network reachability information with BGP systems in other autonomous systems. You use a fully meshed configuration with BGP.

BGP provides routing updates that include a network number, a list of ASs that the routing information passed through, and a list of other path attributes. BGP works with cost metrics to choose the best available path; it sends updated router information only when one host has detected a change, and only the affected part of the routing table is sent.

BGP communicates within one AS using Interior BGP (IBGP) because BGP does not work well with IGP. Thus the routers inside the AS maintain two routing tables: one for the IGP and one for IBGP. BGP uses exterior BGP (EBGP) between different autonomous systems.

BPDU

In STP, a Bridge Protocol Data Unit is a packet that initiates communication between devices. BPDU packets contain information on ports, addresses, priorities, and costs and they ensure that the data ends up where it was intended to go. BPDU messages are exchanged across bridges to detect loops in a network topology. The loops are then removed by shutting down selected bridge interfaces and placing redundant switch ports in a backup, or blocked, state.

bridge

In conventional networking terms, bridging is a Layer 2 function that passes frames between two network segments; these segments have a common network layer address. The bridged frames pass only to those segments connected at a Layer 2 level, which is called a broadcast domain (or VLAN). You must use Layer 3 routing to pass frames between broadcast domains (VLANs).

In wireless technology, bridging refers to forwarding and receiving data between radio interfaces on APs or between clients on the same radio. So, bridged traffic can be forwarded from one AP to another AP without having to pass through the switch on the wired network.

BSS

Basic Service Set is a wireless topology consisting of one access point connected to a wired network and a set of wireless devices. Also called an infrastructure network. See also *IBSS (Independent Basic Service Set)*.

CCM

In CFM, Connectivity Check Messages are frames transmitted periodically by a MEP to ensure connectivity across the maintenance entities to which the transmitting MEP belongs. The CCMs contain a unique ID for the specified domain. Because a failure to receive a CCM indicates a connectivity fault in the network, CCMs proactively check for network connectivity.

CEP

Customer Edge Port, also known as Selective Q-in-Q or C-tagged Service Interface, is a role that is configured in software as a CEP VMAN port, and connects a VMAN to specific CVLANs based on the CVLAN CVID. The CNP role, which is configured as an untagged VMAN port, connects a VMAN to all other port traffic that is not already mapped to the port CEP role.

CFM

Connectivity Fault Management allows an ISP to proactively detect faults in the network for each customer service instance individually and separately. CFM comprises capabilities for detecting, verifying, and isolating connectivity failures in virtual bridged LANs.

CHAP

Challenge-Handshake Authentication Protocol is one of the two main authentication protocols used to verify a user's name and password for PPP Internet connections. CHAP is more secure because it performs a three-way handshake during the initial link establishment between the home and remote machines. It can also repeat the authentication anytime after the link has been established.

CIST root port

In an NSTO environment, the port on the CIST regional root bridge that connects to the CIST root bridge is the CIST root port. The CIST root port is the master port for all MSTIs in that MSTP region, and it is the only port that connects the entire region to the CIST root bridge.

CLI

Command Line Interface. The CLI provides an environment to issue commands to monitor and manage switches and wireless appliances.

CoS

Class of Service specifies the service level for the classified traffic type.

DCB

Data Center Bridging is a set of IEEE 802.1Q extensions to standard Ethernet, that provide an operational framework for unifying Local Area Networks (LAN), Storage Area Networks (SAN) and Inter-Process Communication (IPC) traffic between switches and endpoints onto a single transport layer.

DHCP

Dynamic Host Configuration Protocol allows network administrators to centrally manage and automate the assignment of IP addresses on the corporate network. DHCP sends a new IP address when a computer is plugged into a different place in the network. The protocol supports static or dynamic IP addresses and can dynamically reconfigure networks in which there are more computers than there are available IP addresses.

DoS attack

Denial of Service attacks occur when a critical network or computing resource is overwhelmed so that legitimate requests for service cannot succeed. In its simplest form, a DoS attack is indistinguishable from normal heavy traffic. ExtremeXOS software has configurable parameters that allow you to defeat DoS attacks.

DSCP

Differentiated Services Code Point is a field in the header of IP packets that is used for packet classification purposes.

DSSS

Direct-Sequence Spread Spectrum is a transmission technology used in Local Area Wireless Network (LAWN) transmissions where a data signal at the sending station is combined with a higher data rate bit sequence, or chipping code, that divides the user data according to a spreading ratio. The chipping code is a redundant bit pattern for each bit that is transmitted, which increases the signal's resistance to interference. If one or more bits in the pattern are damaged during transmission, the original data can be recovered due to the redundancy of the transmission. (Compare with *FHSS (Frequency-Hopping Spread Spectrum)*.)

EAP-TLS/EAP-TTLS

EAP-TLS Extensible Authentication Protocol - Transport Layer Security. A general protocol for authentication that also supports multiple authentication methods, such as token cards, Kerberos, one-time passwords, certificates, public key authentication and smart cards.

IEEE 802.1x specifies how EAP should be encapsulated in LAN frames.
In wireless communications using EAP, a user requests connection to a WLAN through an access point, which then requests the identity of the user and transmits that identity to an authentication server such as RADIUS The server asks the access point for proof of identity, which the access point gets from the user and then sends back to the server to complete the authentication.

EAP-TLS provides for certificate-based and mutual authentication of the client and the network. It relies on client-side and server-side certificates to perform authentication and can be used to dynamically generate user-based and session-based WEP keys.

EAP-TTLS (Tunneled Transport Layer Security) is an extension of EAP-TLS to provide certificate-based, mutual authentication of the client and network through an encrypted tunnel, as well as to generate dynamic, per-user, per-session WEP keys. Unlike EAP-TLS, EAP-TTLS requires only server-side certificates.
(See also *PEAP (Protected Extensible Authentication Protocol)*.)

EAPS

Extreme Automatic Protection Switching is an Extreme Networks-proprietary version of the Ethernet Automatic Protection Switching protocol that prevents looping Layer 2 of the network. This feature is discussed in RFC 3619.

ECMP

Equal Cost Multi Paths is a routing algorithm that distributes network traffic across multiple high-bandwidth OSPF, BPG, IS-IS, and static routes to increase performance. The Extreme Networks implementation supports multiple equal cost paths between points and divides traffic evenly among the available paths.

edge safeguard

Loop prevention and detection on an edge port configured for RSTP. Configuring edge safeguard on RSTP edge ports can prevent accidental or deliberate misconfigurations (loops) resulting from connecting two edge ports together or from connecting a hub or other non-STP switch to an edge port. Edge safeguard also limits the impact of broadcast storms that might occur on edge ports. This advanced loop

prevention mechanism improves network resiliency but does not interfere with the rapid convergence of edge ports.

EDP

Extreme Discovery Protocol is a protocol used to gather topology information about neighboring Extreme Networks switches.

ELRP

Extreme Loop Recovery Protocol is an Extreme Networks-proprietary protocol that allows you to detect Layer 2 loops.

ERPS

Ethernet Ring Protection Switching provides fast protection and recovery switching for Ethernet traffic in a ring topology. It also ensures that the Ethernet layer remains loop-free. It is defined in ITU/T G.8032.

ESRP

Extreme Standby Router Protocol is an Extreme Networks-proprietary protocol that provides redundant Layer 2 and routing services to users.

ESRP group

An ESRP group runs multiple instances of ESRP within the same VLAN (or broadcast domain). To provide redundancy at each tier, use a pair of ESRP switches on the group. See *ESRP (Extreme Standby Router Protocol)*.

event

Any type of occurrence on a switch that could generate a log message or require an action. For more, see *syslog*.

Extreme Defender for IoT

Extreme Defender for IoT provides unique in-line security for mission critical and/or vulnerable IoT devices. Placed between the IoT device and the network, the Defender for IoT solution helps secure and isolate IoT devices protecting them from internal and external hacking attempts, viruses, malware and ransomware, DDoS attacks, and more. Designed to be simple and flexible, Defender for IoT can be deployed over any network infrastructure to enable secure IoT management without significant network changes.

The solution is comprised of the Extreme Defender Application Software and the Defender Adapter (SA201) or AP3912i access point. ExtremeCloud IQ Controller is the supported platform for the Extreme Defender Application.

For more information, see https://www.extremenetworks.com/products/wi-fi-management/extremecloud-iq-controller/extremecloud-iq-controller.

ExtremeAnalytics for ExtremeCloud IQ - Site Engine

ExtremeAnalytics™ for ExtremeCloud™ IQ - Site Engine, formerly Purview™, is a network powered application analytics and optimization solution that captures and analyzes context-based application traffic to deliver meaningful intelligence about applications, users, locations, and devices. ExtremeAnalytics for ExtremeCloud IQ - Site Engine provides data to show how applications are being used. This can

be used to better understand customer behavior on the network, identify the level of user engagement, and assure business application delivery to optimize the user experience. The software also provides visibility into network and application performance allowing IT to pinpoint and resolve performance issues in the infrastructure whether they are caused by the network, application, or server. Learn more at https://www.extremenetworks.com/product/extremeanalytics-for-extremecloud-iq-site-engine/.

ExtremeCloud IQ Controller

Extreme Campus Controller has been rebranded to ExtremeCloud IQ Controller. The new ExtremeCloud IQ Controller supports Campus/Centralized sites only.

The ExtremeCloud IQ Controller is a next generation orchestration application offering all the mobility services required for modern unified access deployments offering simplified integration with ExtremeCloud IQ to take advantage of Cloud Visibility into the network. The ExtremeCloud IQ Controller extends the simplified workflows of the ExtremeCloud public cloud application to on-prem/private cloud deployments.

The ExtremeCloud IQ Controller includes comprehensive critical network services for wireless and wired connectivity, wireless device secure onboarding, distributed and centralized data paths, role-based access control through the Application Layer, integrated location services, and IoT device onboarding through a single platform.

Built on architecture with the latest technology, the embedded operating system supports application containers that enable future expansion of value added applications for the unified access edge.

ExtremeCloud IQ - Site Engine

ExtremeCloud™ IQ - Site Engine (formerly known as Extreme Management Center and Netsight), is a web-based control interface that provides centralized visibility into your network. ExtremeCloud IQ - Site Engine reaches beyond ports, VLANs, and SSIDs and provides detailed control of individual users, applications, and protocols. When coupled with wireless and Identity & Access Management products, ExtremeCloud IQ - Site Engine becomes the central location for monitoring and managing all the components in the infrastructure. Learn more at https://www.extremenetworks.com/product/extremecloud-iq-site-engine/.

ExtremeCloud™ IQ

ExtremeCloud™ IQ is an industry-leading and visionary approach to cloud-managed networking, built from the ground up to take full advantage of the Extreme Networks end-to-end networking solutions. ExtremeCloud IQ delivers unified, full-stack management of wireless access points, switches, and routers and enables onboarding, configuration, monitoring, troubleshooting, reporting, and more. Using innovative machine learning and artificial intelligence technologies, ExtremeCloud IQ analyzes and interprets millions of network and user data points, from the network edge to the data center, to power actionable business and IT insights, and deliver new levels of network automation and intelligence. Learn more about ExtremeCloud IQ at https://www.extremenetworks.com/support/documentation/extremecloud-iq/.

ExtremeControl for ExtremeCloud IQ - Site Engine

ExtremeControl for ExtremeCloud™ IQ - Site Engine, formerly Extreme Access Control™ (EAC), is a set of management software tools that use information gathered by a hardware engine to control policy to all devices on the network. The software allows you to automate and secure access for all devices on the network from a central dashboard, making it easier to roll out security and identity policies across the wired and wireless network. Learn more at https://www.extremenetworks.com/product/extremecontrol-for-extremecloud-iq-site-engine/.

ExtremeWireless

ExtremeWireless products and solutions offer high-density Wi-Fi access, connecting your organization with employees, partners, and customers everywhere they go. The family of wireless products and solutions includes APs, wireless appliances, and software. Learn more about ExtremeWireless at http://www.extremenetworks.com/products/wireless/.

ExtremeXOS

ExtremeXOS, a modular switch operating system, is designed from the ground up to meet the needs of large cloud and private data centers, service providers, converged enterprise edge networks, and everything in between. Based on a resilient architecture and protocols, ExtremeXOS supports network virtualization and standards-based SDN capabilities like VXLAN gateway and OpenStack Cloud orchestration. ExtremeXOS also supports comprehensive role-based policy.

FDB

The switch maintains a database of all MAC address received on all of its ports and uses this information to decide whether a frame should be forwarded or filtered. Each forwarding database (FDB) entry consists of the MAC address of the sending device, an identifier for the port on which the frame was received, and an identifier for the VLAN to which the device belongs. Frames destined for devices that are not currently in the FDB are flooded to all members of the VLAN. For some types of entries, you configure the time it takes for the specific entry to age out of the FDB.

FHSS

Frequency-Hopping Spread Spectrum is a transmission technology used in Local Area Wireless Network (LAWN) transmissions where the data signal is modulated with a narrowband carrier signal that 'hops' in a random but predictable sequence from frequency to frequency as a function of time over a wide band of frequencies. This technique reduces interference. If synchronized properly, a single logical channel is maintained. (Compare with *DSSS (Direct-Sequence Spread Spectrum)*.)

gratuitous ARP

When a host sends an ARP request to resolve its own IP address, it is called gratuitous ARP.

IBSS

An IBSS is the 802.11 term for an ad hoc network. See *ad hoc mode*.

ICMP

Internet Control Message Protocol is the part of the TCP/IP protocol that allows generation of error messages, test packets, and operating messages. For example, the ping command allows you to send ICMP echo messages to a remote IP device to test for connectivity. ICMP also supports traceroute, which identifies intermediate hops between a given source and destination.

IGMP

Hosts use Internet Group Management Protocol to inform local routers of their membership in multicast groups. Multicasting allows one computer on the Internet to send content to multiple other computers that have identified themselves as interested in receiving the originating computer's content. When all hosts leave a group, the router no longer forwards packets that arrive for the multicast group.

IGP

Interior Gateway Protocol refers to any protocol used to exchange routing information within an AS. Examples of Internet IGPs include RIP and OSPF.

LAG

A Link Aggregation Group is the logical high-bandwidth link that results from grouping multiple network links in link aggregation (or load sharing). You can configure static LAGs or dynamic LAGs (using the LACP).

LLDP

Link Layer Discovery Protocol conforms to IEEE 802.1ab and is a neighbor discovery protocol. Each LLDP-enabled device transmits information to its neighbors, including chassis and port identification, system name and description, VLAN names, and other selected networking information. The protocol also specifies timing intervals in order to ensure current information is being transmitted and received.

LSDB

In OSPF, a Link State Database is a database of information about the link state of the network. Two neighboring routers consider themselves to be adjacent only if their LSDBs are synchronized. All routing information is exchanged only between adjacent routers.

MD5

Message-Digest algorithm is a hash function that is commonly used to generate a 128-bit hash value. It was designed by Ron Rivest in 1991. MD5 is officially defined in RFC 1321 - The MD5 Message-Digest Algorithm.

MEP

In MEP, a Maintenance End Point is an end point for a single domain, or maintenance association. The MEP may be either an UP MEP or a DOWN MEP.

MIB

Management Information Bases make up a database of information (for example, traffic statistics and port settings) that the switch makes available to network management systems. MIB names identify objects that can be managed in a network and contain information about the objects. MIBs provide a means to configure

a network device and obtain network statistics gathered by the device. Standard, minimal MIBs have been defined, and vendors often have private enterprise MIBs.

MIC

Message Integrity Check (or Code), also called 'Michael', is part of WPA and TKIP. The MIC is an additional 8-byte code inserted before the standard 4-byte ICV appended in by standard WEP to the 802.11 message. This greatly increases the difficulty in carrying out forgery attacks.
Both integrity check mechanisms are calculated by the receiver and compared against the values sent by the sender in the frame. If the values match, there is assurance that the message has not been tampered with.

MLAG

The Multi-switch Link Aggregation Group feature allows users to combine ports on two switches to form a single logical connection to another network device. The other network device can be either a server or a switch that is separately configured with a regular LAG (or appropriate server port teaming) to form the port aggregation.

MPLS

Multiprotocol Label Switching speeds up network traffic. When forwarding packets, the Layer 2 (Switching) label is used to avoid complex destination lookups in the routing table. MPLS uses Label Switched Paths (LSPs) to establish the network path. The packet will be labeled so that service providers can decide the best way to keep traffic flowing. The Multiprotocol Label Switching Transport Profile (MPLS-TP) extensions to MPLS are designed to meet service provider requirements and are used as a network layer technology in transport networks. MPLS-TP gives service providers a reliable packet-based technology that is based on circuit-based transport networking. MPLS-TP is expected to be a low cost level 2 technology (if the limited profile is implemented in isolation) that will provide QoS, end-to-end OAM and protection switching.

MSDP

Multicast Source Discovery Protocol is used to connect multiple multicast routing domains. MSDP advertises multicast sources across Protocol Independent Multicast-Sparse Mode (PIM-SM) multicast domains or Rendezvous Points (RPs). In turn, these RPs run MSDP over TCP to discover multicast sources in other domains.

MSTI

Multiple Spanning Tree Instances control the topology inside an MSTP region. An MSTI is a spanning tree domain that operates within a region and is bounded by that region; and MSTI does not exchange BPDUs or send notifications to other regions. You can map multiple VLANs to an MSTI; however, each VLAN can belong to only one MSTI. You can configure up to 64 MSTIs in an MSTP region.

MSTP

Multiple Spanning Tree Protocol, based on IEEE 802.1Q-2003 (formerly known as IEEE 892.1s), allows you to bundle multiple VLANs into one STP topology, which also provides enhanced loop protection and better scaling. MSTP uses RSTP as the converging algorithm and is compatible with legacy STP protocols.

NetLogin

Network login provides extra security to the network by assigning addresses only to those users who are properly authenticated. You can use web-based, MAC-based, or IEEE 802.1X-based authentication with network login. The two modes of operation are campus mode and ISP mode.

netmask

A netmask is a string of 0s and 1s that mask, or screen out, the network part of an IP address, so that only the host computer part of the address remains. A frequently-used netmask is 255.255.255.0, used for a Class C subnet (one with up to 255 host computers). The ".0" in the netmask allows the specific host computer address to be visible.

OSPF

An interior gateway routing protocol for TCP/IP networks, Open Shortest Path First uses a link state routing algorithm that calculates routes for packets based on a number of factors, including least hops, speed of transmission lines, and congestion delays. You can also configure certain cost metrics for the algorithm. This protocol is more efficient and scalable than vector-distance routing protocols. OSPF features include least-cost routing, ECMP routing, and load balancing. Although OSPF requires CPU power and memory space, it results in smaller, less frequent router table updates throughout the network. This protocol is more efficient and scalable than vector-distance routing protocols.

OSPFv3

Open Shortest Path First version 3 is one of the routing protocols used with IPV6 and is similar to OSPF.

PEAP

Protected Extensible Authentication Protocol is an IETF draft standard to authenticate wireless LAN clients without requiring them to have certificates. In PEAP authentication, first the user authenticates the authentication server, then the authentication server authenticates the user. If the first phase is successful, the user is then authenticated over the SSL tunnel created in phase one using EAP-Generic Token Card (EAP-GTC) or Microsoft Challenged Handshake Protocol Version 2 (MSCHAP V2). (See also *EAP-TLS/EAP-TTLS*.)

PoE

The Power over Ethernet standard (IEEE 802.3af) defines how power can be provided to network devices over existing Ethernet connections, eliminating the need for additional external power supplies.

QoS

Quality of Service is a technique that is used to manage network resources and guarantee a bandwidth relationship between individual applications or protocols. A communications network transports a multitude of applications and data, including high-quality video and delay-sensitive data such as real-time voice. Networks must provide secure, predictable, measurable, and sometimes guaranteed services. Achieving the required QoS becomes the secret to a successful end-to-end business solution.

RADIUS

RADIUS is a client/server protocol and software that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service. RADIUS allows a company to maintain user profiles in a central database that all remote servers can share. It provides better security, allowing a company to set up a policy that can be applied at a single administered network point. With RADIUS, you can track usage for billing and for keeping network statistics.

rate limiting

In QoS, rate limiting is the process of restricting traffic to a peak rate (PR).

rate shaping

In QoS, rate shaping is the process of reshaping traffic throughput to give preference to higher priority traffic or to buffer traffic until forwarding resources become available.

RIP

This IGP vector-distance routing protocol is part of the TCP/IP suite and maintains tables of all known destinations and the number of hops required to reach each. Using Routing Information Protocol, routers periodically exchange entire routing tables. RIP is suitable for use only as an IGP.

RIPng

Routing Information Protocol Next Generation is one of the routing protocols used with IPv6 and is similar to RIP.

RSTP

Rapid Spanning Tree Protocol, described in IEEE 802.1w, is an enhanced version of STP that provides faster convergence. The Extreme Networks implementation of RSTP allows seamless interoperability with legacy STP.

segment

In Ethernet networks, a section of a network that is bounded by bridges, routers, or switches. Dividing a LAN segment into multiple smaller segments is one of the most common ways of increasing available bandwidth on the LAN.

SNMP

Simple Network Management Protocol is a standard that uses a common software agent to remotely monitor and set network configuration and runtime parameters. SNMP operates in a multivendor environment, and the agent uses MIBs, which define what information is available from any manageable network device. You can also set traps using SNMP, which send notifications of network events to the system log.

SNTP

Simple Network Time Protocol is used to synchronize the system clocks throughout the network. An extension of NTP, SNTP can usually operate with a single server and allows for IPv6 addressing.

SSH

Secure Shell, sometimes known as Secure Socket Shell, is a UNIX-based command interface and protocol of securely gaining access to a remote computer. With SSH commands, both ends of the client/server connection are authenticated using a digital certificate, and passwords are protected by being encrypted. At Extreme Networks, the SSH is a separate software module, which must be downloaded separately. (SSH is bundled with SSL in the software module.)

SSL

Secure Socket Layer is a protocol for transmitting private documents using the Internet. SSL works by using a public key to encrypt data that is transferred over the SSL connection. SSL uses the public-and-private key encryption system, which includes the use of a digital certificate.

STP

Spanning Tree Protocol, defined in IEEE 802.1d, used to eliminate redundant data paths and to increase network efficiency. STP allows a network to have a topology that contains physical loops; it operates in bridges and switches. STP opens certain paths to create a tree topology, thereby preventing packets from looping endlessly on the network. To establish path redundancy, STP creates a tree that spans all of the switches in an extended network, forcing redundant paths into a standby, or blocked, state.

STP allows only one active path at a time between any two network devices (this prevents the loops) but establishes the redundant links as a backup if the initial link should fail. If STP costs change, or if one network segment in the STP becomes unreachable, the spanning tree algorithm reconfigures the STP topology and re-establishes the link by activating the standby path.

STPD

Spanning Tree Domain is an STP instance that contains one or more VLANs. The switch can run multiple STPDs, and each STPD has its own root bridge and active path. In the Extreme Networks implementation of STPD, each domain has a carrier VLAN (for carrying STP information) and one or more protected VLANs (for carrying the data).

syslog

A protocol used for the transmission of event notification messages across networks, originally developed on the University of California Berkeley Software Distribution (BSD) TCP/IP system implementations, and now embedded in many other operating systems and networked devices. A device generates a messages, a relay receives and forwards the messages, and a collector (a syslog server) receives the messages without relaying them.
syslog uses the UDP as its underlying transport layer mechanism. The UDP port that has been assigned to syslog is 514. (RFC 3164)

UDP

User Datagram Protocol is an efficient but unreliable, connectionless protocol that is layered over IP (as is TCP). Application programs must supplement the protocol to provide error processing and retransmitting data. UDP is an OSI Layer 4 protocol.

VLAN

The term VLAN is used to refer to a collection of devices that communicate as if they are on the same physical LAN. Any set of ports (including all ports on the switch) is considered a VLAN. LAN segments are not restricted by the hardware that physically connects them. The segments are defined by flexible user groups you create with the CLI.

VMAN

In ExtremeXOS software, Virtual MANs are a bi-directional virtual data connection that creates a private path through the public network. One VMAN is completely isolated from other VMANs; the encapsulation allows the VMAN traffic to be switched over Layer 2 infrastructure. You implement VMAN using an additional 892.1Q tag and a configurable EtherType; this feature is also known as Q-in-Q switching.

VR-Control

This virtual router is part of the embedded system in Extreme Networks switches. VR-Control is used for internal communications between all the modules and subsystems in the switch. It has no ports, and you cannot assign any ports to it. It also cannot be associated with VLANs or routing protocols. (Referred to as VR-1 in earlier ExtremeXOS software versions.)

VR-Default

This virtual router is part of the embedded system in Extreme Networks switches. VR-Default is the default VR on the system. All data ports in the switch are assigned to this VR by default; you can add and delete ports from this VR. Likewise, VR-Default contains the default VLAN. Although you cannot delete the default VLAN from VR-Default, you can add and delete any user-created VLANs. One instance of each routing protocol is spawned for this VR, and they cannot be deleted. (Referred to as VR-2 in earlier ExtremeXOS software versions.)

VR-Mgmt

This virtual router is part of the embedded system in Extreme Networks switches. VR-Mgmt enables remote management stations to access the switch through Telnet, SSH, or SNMP sessions; and it owns the management port. The management port cannot be deleted from this VR, and no other ports can be added. The Mgmt VLAN is created VR-Mgmt, and it cannot be deleted; you cannot add or delete any other VLANs or any routing protocols to this VR. (Referred to as VR-0 in earlier ExtremeXOS software versions.)

VRRP

The Virtual Router Redundancy Protocol specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IP address(es) associated with a virtual router is called the master router, and forwards packets sent to these IP addresses. The election process provides dynamic failover in the forwarding responsibility should the master router become unavailable. In case the master router fails, the virtual IP address is mapped to a backup router's IP address; this backup becomes the master router. This allows any of the virtual router IP addresses on the LAN to be used as the default first-hop router by end-hosts. The advantage gained from using VRRP is a higher availability default

path without requiring configuration of dynamic routing or router discovery protocols on every host. VRRP is defined in RFC 2338.

VXLAN

VXLAN is a network virtualization technology that leverages existing Layer 3 infrastructures to create tenant overlay networks. VXLAN addresses the requirements of a multi-tenant data center network infrastructure by:

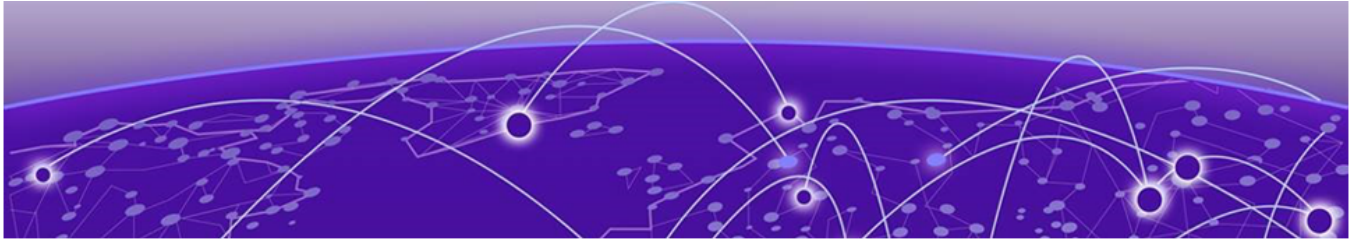
- Increasing virtual network scalability to 16 million instances. This allows for tenant VLAN (Virtual LAN) isolation whereby multiple tenants can manage their own VLAN/VMAN and MAC address spaces.
- Adding an encapsulation that effectively hides VM MAC addresses from the physical network that results in physical networking devices to have smaller MAC and IP table sizes.
- Allowing for Layer 2 adjacency across IP networks by which DC network operators protect their investment in their current infranstructure. Additionally operators can distribute traffic loads across links efficiently using Layer 3 ECMP.

WLAN

Wireless Local Area Network.

XNV

Extreme Network Virtualization is an ExtremeXOS feature that enables the software to support VM port movement, port configuration, and inventory on network switches.



ping

```
ping {count count {start-size start-size} | continuous {start-size
   start-size} | {start-size start-size {end-size end-size}}} {udp}
   {dont-fragment} {ttl ttl} {tos tos} {interval interval} {vr vrid}
   {ipv4 host | ipv6 host} {from} {with record-route}
```

Description

Enables you to send User Datagram Protocol (UDP) or *ICMP (Internet Control Message Protocol)* echo messages or to a remote IP device.

Syntax Description

count	Specifies the number of ping requests to send.
start-size	Specifies the size, in bytes, of the packet to be sent, or the starting size if incremental packets are to be sent.
continuous	Specifies that UDP or ICMP echo messages to be sent continuously. This option can be interrupted by pressing **[Ctrl]** + **C**.
end-size	Specifies an end size for packets to be sent.
udp	Specifies that the ping request should use UDP instead of ICMP.
dont-fragment	Sets the IP to not fragment the bit.
ttl	Sets the TTL value.
tos	Sets the TOS value.
interval	Sets the time interval between sending out ping requests.
vr	Specifies the virtual route to use for sending out the echo message. If not specified, *VR-Default* is used. **Note:** User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.
ipv4	Specifies IPv4 transport.
ipv6	Specifies IPv6 transport. **Note:** If you are contacting an IPv6 link local address, you must specify the *VLAN (Virtual LAN)* you are sending the message from: `ping` **`ipv6`** *`link-local address %vlan_name host`*.

host	Specifies a host name or IP address (either v4 or v6).
from	Uses the specified source address. If not specified, the address of the transmitting interface is used.
with record-route	Sets the traceroute information.

Default

N/A.

Usage Guidelines

The `ping` command is used to test for connectivity to a specific host.

You use the ipv6 variable to ping an IPv6 host by generating an ICMPv6 echo request message and sending the message to the specified address. If you are contacting an IPv6 link local address, you must specify the VLAN you are sending the message from, as shown in the following example (you must include the % sign):

`ping` **`ipv6`** *`link-local address %vlan_name host`*.

The `ping` command is available for both the user and administrator privilege level.

When the IPv6 host ping fails, the following error message appears:

`Error: cannot determine outgoing interface. Link local address must be of form LLA% vlan_name.`

Due to upgrading ExtremeXOS 30.1 to 4.14 Linux kernel, ping success to local IP addresses does not depend on link-layer status. Earlier releases of ExtremeXOS had customized Linux behavior that meant that pinging a local VLAN interface would fail when the local interface was down. However, in ExtremeXOS 30.1, pinging a local VLAN interface that is down will result in a successful ping.

If you have an asymmetric routing to a specific destination (where the traffic from the source to the destination uses one path, and the return traffic uses another), use this command with option 7, **`with record-route`**.

For example:

```
ping <destination address> with record-route
ping <destination address> from <source-address> with record-route
```

For more information about this option, see `enable ip-option record-route`.

Example

The following example enables continuous ICMP echo messages to be sent to a remote host:

```
ping continuous 123.45.67.8
```

The following example uses the with record-route option:

```
ping 10.2.1.1 from 10.2.1.2 with record-route
Ping(ICMP) 10.2.1.1: 4 packets, 8 data bytes, interval 1 second(s).
16 bytes from 10.2.1.1: icmp_seq=0 ttl=63 time=10 ms
RR: 10.2.1.2    10.2.0.21    10.2.1.1    10.2.1.1    10.2.0.1    10.2.1.2
```

History

This command was first available in ExtremeXOS 10.1.

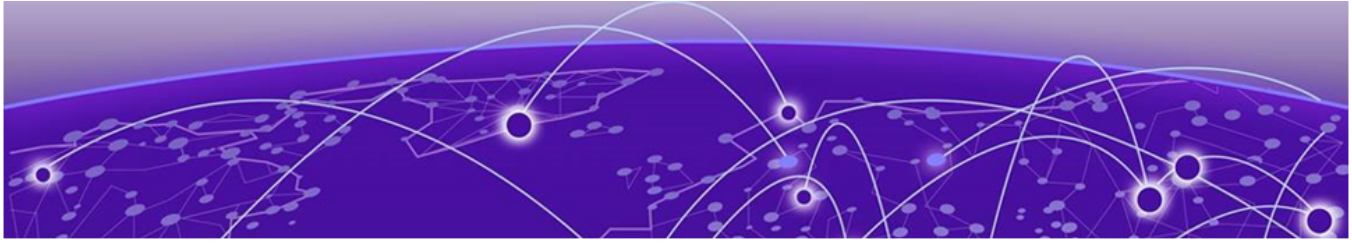
The **ipv6** option was added in ExtremeXOS 11.2.

The IPv6 error message was modified in ExtremeXOS 15.2.

Ping success to local IP addresses not depend on link-layer status added in ExtremeXOS 30.1.

Platform Availability

This command is available on all Universal switches supported in this document.



telnet

```
telnet {vr vr_name} [host_name | remote_ip] {port}
```

Description

Allows you to Telnet from the current command-line interface session to another host.

Syntax Description

vr	Specifies use of a virtual router. **Note:** User-created VRs are supported only on the platforms listed for this feature in the Switch Engine v33.6.1 Licensing Guide document.
vr_name	Specifies the name of the virtual router.
host_name	Specifies the name of the host.
remote_ip	Specifies the IP address of the host.
port	Specifies a TCP port number. The default is port 23.

Default

- Telnet—enabled.
- Virtual router—Uses all virtual routers on the switch for outgoing Telnet requests.
- Port—23.

Usage Guidelines

Only VT100 emulation is supported.

Before you can start an outgoing Telnet session, you need to configure the switch IP parameters. To open a Telnet connection, you must specify the host IP address or the host name of the device you want to connect to. Check the user manual supplied with the Telnet facility if you are unsure of how to do this. Although the switch accepts IPv6 connections, you can only Telnet from the switch to another device with an IPv4 address.

You must configure DNS in order to use the host_name option.

Host Name and Remote IP Address Character Restrictions

This section provides information about the characters supported by the switch for host names and remote IP addresses.

When specifying a host name or remote IP address, the switch permits only the following characters:

- Alphabetical letters, upper case and lower case (A-Z, a-z)
- Numerals (0-9)
- Period (.)
- Dash (-) Permitted only for host names
- Underscore (_) Permitted only for host names
- Colon (:)

When naming or configuring an IP address for your network server, remember the requirements listed above.

Virtual Router Requirements

The vr_name option specifies the name of the virtual router. The valid virtual router names at system boot-up are *VR-Mgmt*, *VR-Control*, and *VR-Default*; however, you can only Telnet on VR-Mgmt and VR-Default. In ExtremeXOS 10.1, the valid virtual routers are VR-0, VR-1, and VR-2 respectively, and Telnet used VR-0 by default. For more information about virtual routers, see Virtual Routers.

Example

The following command starts a Telnet client communication to the host at IP address 123.45.67.8:

```
telnet 123.45.67.8
```

The following command starts a Telnet client communication with a host named sales:

```
telnet sales
```

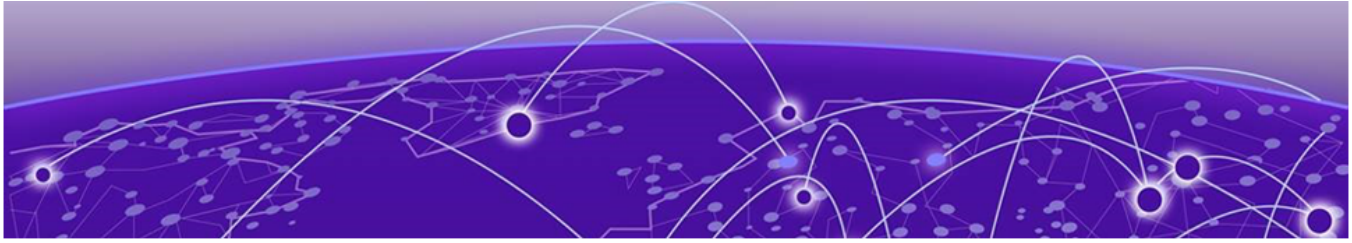
History

This command was first available in ExtremeXOS 10.1.

Support for the following virtual routers was added in ExtremeXOS 11.0: VR-Mgmt and VR-Default.

Platform Availability

This command is available on all Universal switches supported in this document.



virtual-router

```
virtual-router {vr-name}
```

Description

Changes the VR context.

Syntax Description

vr-name	Specifies the name of the VR.

Default

N/A.

Usage Guidelines

Use this command to change the VR context for subsequent commands. When you issue the command, the prompt changes to reflect the VR domain. Configuration commands for Layer 3 routing protocols, creating VLANs, and deleting VLANs apply only to the current VR context.

Use this command with no name, or use the name "*VR-Default*" to return to the default configuration domain.

Under a VR configuration domain, any VR commands are applied only to that VR. The VR commands consist of all the *BGP (Border Gateway Protocol)*, *OSPF (Open Shortest Path First)*, PIM, and *RIP (Routing Information Protocol)* commands, and the commands listed in the following table.

Table 180: VR Commands

`[enable | disable] ipforwarding`
clear iparp
`clear counters iparp`[12]
`configure iparp`[12]
`configure iparp [add | delete]`[12]
`[enable | disable] iparp`[12]

[12] Other commands are available with these listed.

Table 180: VR Commands (continued)

`show iparp` [12]
`configure iproute [add | delete]` [12]
`show iproute` [12]
`show ipstats` [12]
`rtlookup`
`create [vlan | vman] vlan-name`
`[enable | disable] igmp`
`[enable | disable] igmp snooping` [12]
`[enable | disable] ipmcforwarding`
`show igmp`
`show igmp snooping`
`show igmp group`
`show igmp snooping cache`
`[enable | disable] mld`
`[enable | disable] mld snooping`
`show mld`
`show mld snooping`
`show mld group`

The VR context simplifies configuration because you do not have to specify the VR for each individual protocol configuration command. The current VR context is indicated in the CLI prompt.

For example, if you wish to configure OSPF for the user VR vr-manufacturing, you would change the VR context to that of vr-manufacturing. All the subsequent OSPF commands would apply to that VR, unless the context is changed again.

A VR is identified by a name (up to 32 characters long). The name must be unique among the *VLAN (Virtual LAN)* and VR names on the switch. For backward compatibility, you cannot name a virtual router VR-0, VR-1, or VR-2. VR names are case insensitive.

When a new VR is created, by default, no ports are assigned, no VLAN interface is created, and no support for any routing protocols is added.

Example

The following example changes the VR context to "vr-acme":

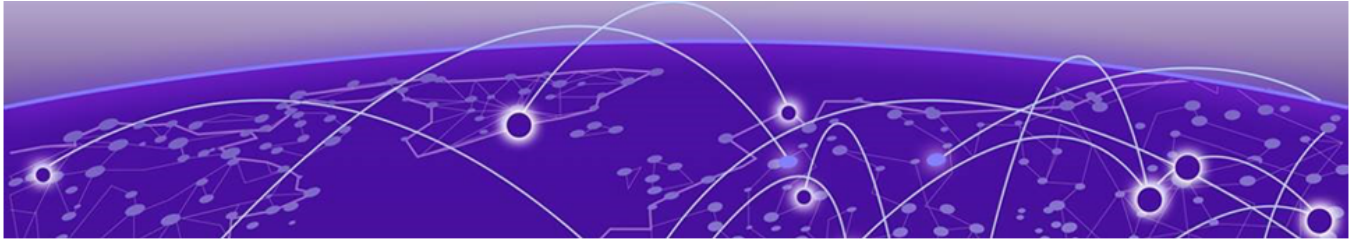
```
virtual-router vr-acme
```

History

This command was first available in ExtremeXOS 11.0.

Platform Availability

This command is available on all Universal switches supported in this document.



Index

Specials

.cfg file 1712
.gz file 2128
.pol file 1449
.xmod file 1711
.xos file 1711

Numerics

10 gigabit ports 238
802.1ad 2261
802.1D 1787
802.1D-2004 1787
802.1p
    default map to egress QoS profiles 1470
    examination feature 1458
    priority replacement 1477
    traffic groups 1458
802.1Q
    amended for vMANs 2261
    encapsulation, TLS 1070
    tagging 2217, 2218
802.1Q-2003 1829
802.1s 1829
802.1w 1817
802.1X
    and NAP 1242
802.1X authentication
    advantages 1221
    co-existence with web-based 1221
    configuration, example 1238
    disadvantages 1221
    interoperability requirements 1237
    methods 1236
    requirements 1220
    VLAN movement, post-authentication 1242
802.3af 1429

A

access levels 50
account types
    admin 51
    user 51
accounts
    creating 54
    default 54
    deleting 54
    failsafe 55
accounts.
    viewing 54
ACL-based traffic groups 1458
ACLs
    .pol file 1449
    action modifiers 351
    actions 350
    byte counters 369
    counters 385
    description 341
    dynamic 370, 1332
    editing 1449
    egress 351
    examples 385, 387
    match conditions 355
    metering 1469
    packet counter 369
    priority 373
    refreshing 1451
    rule entry 346
    rule syntax 346
    slices 388
    smart refresh 1451
    transferring to the switch 1449
    troubleshooting 1449, 1469
acquired node
    definition 158
action modifiers
    ACL 351
action statements, policy 1525
actions
    ACL 350
Active Directory 2153
active interfaces 1163
active node 157
active topology 157
Address Resolution Protocol, *see* ARP
address-based load-sharing 268
admin account 54
Adspec 1091
advertisement interval, EDP 319
advertising labels 1060, 1061
agent, local 1773
agent, RMON 1777
aging entries, FDB 827
alarm actions 1781
alternate IP address 192
alternate stacking
    ports 150

announcements 8
area 0
  OSPFv3 1398, 1419
areas
  OSPFv3 1398, 1419
ARP
  and VLAN aggregation 932
  communicating with devices outside subnet 916
  configuring proxy ARP 914
  disabling additions on superVLAN 933
  gratuitous ARP protection 1579
  incapable device 914
  learning
    adding permanent entries 1577
    configuring 1577
    DHCP secured ARP 1577
    displaying information 1578
    overview 1576
  proxy ARP between subnets 916
  proxy ARP, description of 914
  responding to ARP requests 914
  validation
    configuring 1581
    displaying information 1582
AS
  BGP private numbers 434
  description
    BGP 421
    IS-IS 992
    OSPF 1391
    OSPFv3 1413
  expressions 1523
ASCII-formatted configuration file
  downloading 1717
  loading 1717
  support 139
  troubleshooting 1716
  uploading 1716
  verifying 1717
authentication
  local database 1232
authentication methods
  802.1X 1236
  MAC-based 1255
  web-based 1246
AuthnoPriv 121
AuthPriv 121
auto-negotiation
  description 225, 227–229
  flow control 225, 227–229
  on 227
  support 227
autobind ports 1807
automatic restart, ELSM 1744
autonegotiation
  displaying setting 331
autonomous system. , *see* AS
autopolarity 239

B

backbone area
  OSPFv3 1398, 1419
backup node
  configuring 169
  definition 157
  redundancy 147
banner
  string 46
  warning 52
base URL, network login 1247
BFD 1309
BGP
  and IP multinetting 919
  attributes 422
  autonomous system
    description 421
    path 422
  cluster 427
  community 423
  description 421
  examples
    route confederations 429, 463
    route reflector 428, 459
  loopback interface 430
  peer groups
    creating 431
    deleting 431
    description 431
    mandatory parameters 431
    neighbors 448
  private AS numbers 434
  route aggregation
    description 451
  route confederations 429
  route flap dampening
    configuring 447
  route reflectors 427
  route selection 433
  static networks 435
BGP (Border Gateway Protocol) 421, 1149
Bidirectional Forwarding Detection. , *see* BFD
binding labels, description of 1064
blackhole entries, FDB 829, 1548
BOOTP
  relay
    configuring 922
    viewing 925
  server 73
  using 73
BootROM
  displaying 1724
Bootstrap Protocol , *see* BOOTP
Border Gateway Protocol. , *see* BGP
broadcast traffic, translation VLAN 2250
bulk checkpointing 88

C

cabling
  10/10/1000BASE-T ports 239
  crossover cables 239
calculated LSP 1065
calculated LSP next hop
  managing 1067
  matching 1066
campus mode 1225
candidate node 157
CCM
  ping 1288
  traceroute 1288
CFM
  CCM messages 1286
  CFM messages 1286
  configuring CCMs 601, 1293
  configuring domains 1290
  configuring MAs 1290, 1291
  configuring MEPs 600, 1293
  configuring MIPs 600, 1293
  configuring ping 1294
  configuring traceroute 1294
  displaying 1295
  domain format 1290
  domains 1286
  Ethernet types 1283
  example 1295
  implementation 1283
  MA and domain 1286
  MA formats 1291
  MA levels 1286
  MAC addresses 1283
  mapping 1286
  MEPs 1286
  MIPs 1287
  MPs 1286
  number of ports 1286
  troubleshooting 1283
  TTL 1287
  verifying 1295
  VLAN association with MAs 1286
checkpointing
  bulk 88
  dynamic 88
  statistics, displaying 88
CIR 1463
CIST
  BPDUs
    CIST records 1832
    M-records 1832
  configuring 1833
  definition 1832
  enabling 1833
  root port 1833
  see also MSTP
CLEAR-Flow
CLEAR-Flow *(continued)*
  configuring 515
  enabling and disabling 516
  overview 501
  rule types 504
CLI
  * prompt 53
  # prompt 51, 53
  > prompt 51, 53
  access levels 50
  command shortcuts 38
  configuration access 51
  line-editing keys 44
  named components 34
  prompt line 53
  starting up 57
  symbols 42
  syntax 32
  syntax helper 33
  syntax symbols (table) 42
  users
    adding 54
    deleting 54
    viewing 54
  using 32
CLI scripting
  built-in functions 2192
  control structures 2191
  descriptions 2182
  examples 2197
  operators 2190
  special characters 2189
  supported TCL functions 2192
  variables 2187
cluster 427
collector, remote 1773
combination ports 330
combo ports 330
command
  prompts 51, 53
  shortcuts 38
command line interface., *see* CLI
command syntax, understanding 32
committed information rate 1463
Common and Internal Spanning Tree, *see* CIST
common commands (table) 45
communicating with devices outside subnet 916
community strings
  private 109
  public 109
  read 109
  read-write 109
complete sequence number PDU 992
compliant frame delay and delay variance measurement-Y.1731 1296
components, EMS 1755
conditions, EMS 1755
configuration

configuration *(continued)*
command prompt 51, 53
domain, virtual router 2205
logging changes 1765
primary and secondary 1712
viewing current 1714
configuration examples 2252
configuration file
.cfg file 1712
ASCII-formatted 139
copying 137
database 139
deleting 139
description 1712
displaying 137, 139
downloading 139, 1720
file types 139
managing 134, 139
overview 139
relaying from primary to backup 87
renaming 136
saving changes 1712
selecting 1712
uploading 139, 1719
using 1712
XML-formatted 139
configuration mode
XML 139
configuring
LDP session timers 1114
PHP 1111
resetting parameters 1116
stack 164, 168
VPLS domain 1127
connectivity 61
Connectivity Fault Management., *see* CFM
conservative label retention mode 1061
console
connection 69
maximum sessions 67
control path 156
control plane redundancy 159
control structures, CLI scripting 2191
control VLAN, EAPS 562
controlling Telnet access 76
copying
files 137
core dump file
.gz file 2128
copying to the switch 2129
copying to the tftp server 2129
description 2128
sending to the switch 2128
core image , *see* image
CoS-based traffic groups 1458
CPU monitoring
description 1782
disabling 1782
CPU monitoring *(continued)*
enabling 1782
troubleshooting 1782
CPU utilization, history 1782
CPU utilization, TOP command 2130
CSNP 992
customer tag 2262
cut-through switching 245

D

daisy chain topology 149, 213, 218
data port 158
database applications, and QoS 1456
database overflow, OSPF 1395
debug information 2128
debug mode 1765, 2127
DECNet protocol filter 2220, 2221
default
accounts 54
gateway 618, 885, 2276
passwords 57
port status 224
routes 889, 966
users 54
default gateway 885
denial of service protection
configuring 1587
description 1586
disabling 1587
displaying settings 1588
enabling 1587
description 1184
designated intermediate system., *see* DIS
destination VLAN
creating 1233
destination VLAN, network login 1233
device triggers 2138
DHCP
bindings database 1566
disabling 1563
displaying settings 1565
enabling 1563
network login and 1220
relay
and IP multinetting 920
configuring 922
viewing 925
requirement for web-based network login 1220
secured ARP 1577
server
and IP multinetting 920, 1563
configuring 1564
snooping
configuring 1567
disabling 1568
displaying information 1568
overview 1566
trusted ports

DHCP *(continued)*
trusted ports *(continued)*
configuring 1568
overview 1568
trusted server
configuring 1568
displaying information 1568
overview 1566
diagnostics
displaying 1728, 1736
slot 1734
switches 1735
system 1734
DiffServ
code point 1459
examination feature 1459
traffic groups 1459
DIS 994
disabling route advertising
RIP 1510, 1517
dismantling a stack 211
displaying settings 331
distance-vector protocol, description 1508, 1515
DNS
configuring 61
description 61
documentation
feedback 7
Domain Name Service *, see* DNS
domains, CFM 1286
domains, ESRP 625
domains, STP 1801
downloading
ASCII-formatted configuration 1717
configuration 1720
downstream unsolicited (DU), definition of 1058
downstream unsolicited mode 1061
downstream-on-demand mode 1061
DSCP
default map to QoS profiles 1459
replacement 1478
dual master condition 149
dual master situation 214, 215
dual primary situation 213
dual-rate QoS 1463
duplex setting, ports 225, 227–229
duplex, displaying setting 331
dynamic
ACLs 370, 1332
checkpointing 88
FDB entries 827, 1548
hostname 996
MVR 1201
netlogin
dynamic VLANs
description 1263
routes
IPv4 888, 966
dynamic *(continued)*
VLANs 1263
see also netlogin
Dynamic Host Configuration Protocol *, see* DHCP

E

EAPOL and DHCP 1221
EAPS
and IP multinetting 920
and MVR 1206–1208
configuring 561, 569
control VLAN 562
disabling
domain 566
loop protection 568
on a switch 565
EAPS domain
creating and deleting 562
enabling
domain 566
loop protection 568
on a switch 565
failed state 565
failtime expiry action 565
failtimer 565
Fast Convergence 566
health-check packet 564
hellotime 564
hitless failover support 560
loop protection messages 567
master node 563
multiple domains per switch 552
names 34
polling timers, configuring 564
primary port 551, 563
protected VLAN 562
ring port, unconfiguring 567
secondary port 551, 563
shared port
configuration rules 568
configuring the domain ID 570
creating and deleting 569
defining the mode 569
status information, displaying 571, 572
switch mode, defining 563
transit node 563
troubleshooting 563
EAPS (Extreme Automatic Protection Switching) 548
Easy-Setup 159
edge safeguard
description 1820
disabling 1820
enabling 1820
EDP
advertisement interval 319
clearing counters 318
default 318
description 318

EDP *(continued)*
disabling 318
enabling 318
timeout interval 319
viewing information 319
egress ACLs 351
egress flooding
displaying 835
guidelines 834
egress port Qos 1472
egress QoS profiles 1470
election
node role 158
priority 158
election algorithms, ESRP 623
ELRP
and ESRP 634
behavior
ESRP master switch 635
ESRP pre-master switch 635
description 634
loop detection 2118
standalone 2118
without ESRP 2118
ELSM
and Layer 2 protocols 1747
automatic restart 1744
configuration example 1748
configuring
hello timer 1743
hold threshold 1744
description 1737
disabling 1745
displaying information 1745
ELSM link state 1740
enabling 1743
fault detection 1737
hello messages 1738
hello transmit states 1738
hold threshold 1743
link state 1739
port states
down 1738
down-stuck 1739
down-wait 1738
up 1738
sticky threshold 1744
timers
down 1742
hello 1742
hellorx 1742
up 1742
EMISTP
description 1804
example 1812
rules 1814
EMS
and dual MSM systems 1752
EMS *(continued)*
configuring targets
components 1755
conditions 1755
description 1752
severity 1754
subcomponents 1755
debug mode 1765
description 1750
displaying messages
console 1762
session 1762
event message formats 1761
expressions
matching 1758
regular 1759
filtering event messages 1752
filters
configuring 1757
creating 1756
viewing 1757
log target
default 1751
disabling 1751
enabling 1751
types 1751
logs
displaying 1762
displaying counters 1764
uploading 1763
parameters
behavior 1761
matching 1759
trigger, event 2141
viewing components and subcomponents 1755
viewing conditions 1755
encapsulation modes 1804
see also STP
EPICenter support 71
ESRP
802.1Q tag 626
and IP multinetting 920
and load sharing 641
and OSPF 622
and STP 633
and VRRP 633, 2279
auto toggle 627
direct link 628
displaying data 642
domain ID 626
domains, description 625
don't count 641
election algorithms 623
environment tracking 637
ESRP-aware 629
ESRP-aware portlist 642
examples 646
failover time 622

ESRP *(continued)*
groups 626
host attach 640
linking switches 629
load sharing and 641
master
behavior 621
definition 618
determining 620
electing 622
election algorithms 623
multiple VLANs sharing a host port 629
neutral state, behavior 621
ping tracking 638
port restart 639
port weight 627
pre-master
behavior 621
timeout 622
reasons to use 618
restarting ports 639
route table tracking 638
selective forwarding 642
slave mode
behavior 621
troubleshooting 629, 632
VLANid 626
ESRP-aware, description 629
Event Management System , *see* EMS
Events, RMON 1779
examples
disconnecting devices 1554
reconnecting devices 1554
EXP field 1063
explicit packet marking, QoS 1457
explicit route 1096
extended IPv4 host cache feature 897
extended tunnel ID 1094
Extreme Discovery Protocol, *see* EDP
Extreme Loop Recovery Protocol. , *see* ELRP
Extreme Multiple Instance Spanning Tree Protocol, *see* EMISTP

F

failover 86, 158, 159
failsafe account 55
fan tray information 1749
fast path routing 896
fault protection 550
FDB
configuring aging time 831
creating a permanent entry example 830
description 826
dynamic entries
limiting 1548
lock down 1551
egress flooding 833
entries
FDB *(continued)*
entries *(continued)*
adding 826
aging 827
blackhole 829
contents 826
dynamic 827
limiting 832
multicast with multiport entries 830
non-aging 828
non-permanent dynamic entry 828
prioritizing 832
PVLAN 2236
static 828
MAC learning 832
prioritizing entries 1538
FEC
binding labels 1064
definition of 1056, 1058
propagating labels 1060
feedback 7
file syntax, policy 1521
file system administration 134
filename requirements 135, 136, 2130
filenames, troubleshooting 135, 136, 2130
files
copying 137
deleting 139
displaying 137
renaming 136
filters
label advertisement 1114
filters, protocol 2220, 2221
firmware
displaying 1724
fixed filter reservation style 1092
flooding 833
flooding, displaying 331
flow control
displaying setting 331
Gigabit Ethernet ports 225, 227–229
Flow Monitor 239–243
forwarding database , *see* FDB
Forwarding Equivalence Class , *see* FEC
forwarding rules, MVR 1204

G

graceful OSPF restart 1396, 1417
gratuitous ARP
description 1579
enabling 1580
GRE tunnel 1182, 1183
GRE Tunnel 887
Greenwich Mean Time Offsets (table) 129
groups
ESRP 626
SNMPv3 119
guest VLAN

guest VLAN *(continued)*
creating 1241
description 1239
disabling 1241
enabling 1241
guidelines 1240
scenarios 1240
settings 1242
troubleshooting 1240
unconfiguring 1241

H

Health Chidk Link Aggregation 278
helper-mode 1397, 1418
History, RMON 1779
hitless failover
description 90
EAPS 560
network login 1226
protocol support 90
STP 1809
VRRP 2288
hold threshold, ELSM 1743
host attach, ESRP 640
HTTP
disabling 1643
enabling 1643
overview 131, 1643
Hypertext Transfer Protoco , *see* HTTP
Hypertext Transfer Protocol, *see* HTTP

I

IEEE 802.1ad 273
IEEE 802.1D 1787
IEEE 802.1D-2004 1787
IEEE 802.1Q 2217
IEEE 802.1Q-2003 1829
IEEE 802.1s 1829
IEEE 802.1w 1817
IEEE 802.1X 1236
IEEE 802.3af 1429
IGMP
and IP multinetting 919
snooping 1184
snooping filters 1186
static 1187
image
.xos file 1711
downloading 1707
EPICenter, using 1707
primary and secondary 1706
upgrading 1704
version string 1709
implicit NULL labels 1064
in-profile traffic 1462
independent LSP control 1062
inheriting ports, MSTP 1808
interfaces
active 1163
IP multinetting 917
IPv6 router 885, 956
passive 1163
Internet Group Management Protocol , *see* IGMP
Internet Router Discovery Protocol, *see* IRDP
interoperability requirements, 802.1X authentication
1237
IP
alternate 192
for stack 192
fragmentation 265
gateway 192
multicast forwarding, configuring 1190
protocol filter 2220, 2221
security
ARP learning 1576
ARP validation 1581
dependencies 1565
DHCP bindings database 1566
DHCP snooping 1566
gratuitous ARP 1579
source IP lockdown 1574
trusted DHCP server 1566
switch address entry 75
IP multicast routing
description 1158
IGMP
description 1184
snooping filters 1186
PIM mode interoperation 1167
PIM multicast border router (PMBR) 1167
PIM-DM 1164
PIM-SM 1166
IP multinetting
configuring 921
description 916
example 921
interface 917
interoperability with
BGP 919
DHCP relay 920
DHCP server 920, 1563
EAPS 920
ESRP 920
IGMP, IGMP snooping 919
IRDP 918
OSPF 918
PIM 920
RIP 919
STP 920
recommendations 916
topology 917
IP unicast routing
BOOTP relay 922
configuration examples 912, 984
DHCP relay 922

IP unicast routing *(continued)*
enabling 907
multinetting
description 916
example 921
proxy ARP 914
routing table
default routes 889, 966
dynamic routes 888, 966
populating 888
static routes 888, 966
IPv6
displaying VLANs 2229
ping 61
protocol filter 2220, 2221
scoped addresses 958
VLANs 2213, 2227
IPv6 unicast routing
enabling 970
relative priorities 890, 968
router interfaces 885, 956
routing table IPv6
multiple routes 889, 967
populating 965
verifying the configuration 973
IPX protocol filter 2220, 2221
IPX_8022 protocol filter 2220, 2221
IPX_SNAP protocol filter 2220, 2221
IRDP, and IP multinetting 918
IS-IS
authentication 996, 1601
autonomous system 992
broadcast adjacency 994
complete sequence number PDU 992
designated intermediate system 994
dynamic hostname 996
establishing adjacencies 992
hello PDU 992
hierarchy 995
IPv4 and IPv6 topology modes 997
link state database 992
metric types 997
operation with IP Routing 995
overview 992
partial sequence number PDU 992
point-to-point adjacency 992
redistributing routes
configuring 999, 1403, 1424, 1511, 1517
restart feature 997
route leaking 997
isolated subscriber VLAN 2234
ISP mode 1225

J

jumbo frames
configuring MPLS modules 1065
description 263
enabling 264
jumbo frames *(continued)*
IP fragmentation 265
path MTU discovery 265
viewing port settings 331
vMANs 264

K

keys
line-editing 44
port monitoring 1732

L

label
advertising 1060
advertising modes 1061
binding 1064
configuring advertisement filters 1114
definition 1056
locally assigned 1064
object 1094
propagating 1060
remotely assigned 1064
retention modes 1061
stack 1063
swapping, definition 1058
Label Edge Router *, see* LER
Label Switch Path *, see* LSP
LACP*, see* link aggregation
LAG (*, see* link aggregation
latestReceivedEngineTime 112
Layer 1, troubleshooting 2107
Layer 2
protocols and ELSM 1747
Layer 3
PVLAN communications 2238
LDAP 2153
LDP
definition of 1058, 1059
hello-adjacency 1060
message exchange 1060
neighbor discovery protocol 1060
session timers, configuring 1114
LEDs
stack number indicator 147
LER
definition of 1058
LFS
description 238
troubleshooting 238
liberal label retention mode 1061
license mismatch 220
licensing
SummitStack 184
SummitStack level restriction 186
upgrades 187
limit, sFlow maximum CPU sample limit 1775
limiting entries, FDB 832

line-editing keys 44
link aggregation
adding or deleting ports 282
algorithms 268
and control protocols 266
and software-controlled redundant ports 267
broadcast, multicast, and unknown packets 279, 280
description 266
dynamic 268
example 286
LACP
active and standby ports 273
and ELSM 283
configuring 283
defaulted port action 273
LAG 282
master port 282
verifying configuration 284
maximum ports and groups 279, 280
restrictions 280
static 268
troubleshooting 266, 283
see also load sharing
link failure 221
Link Fault Signal , *see* LFS
link state database 992
link types
configuring in MSTP 1819
configuring in RSTP 1819
link-state advertisement , *see* LSA
link-state database , *see* LSDB
link-state protocol, description 1508, 1515
linkaggregation
health check 278
LLDP
and 802.1X 1043
Avaya-Extreme information 1055
avaya-extreme TLVs 1038
clearing entries 1052
collecting supplicant information 2150
EMS messages 1043
ethertype 1041
IP address advertisement 1043
length limit 1041
LLDP-MED
fast start 1040
TLVs 1039
traps 1041
LLDPDU 1041
mandatory TLVs 1036
MED information 1055
multicast address 1041
neighbor information 1055
port configuration information 1054
restoring defaults 1053
SNMP traps 1043
statistics 1054
LLDP *(continued)*
supplicant configuration parameters 2151
system description TLV 1046
timers 1043
troubleshooting 1041, 1050, 1051
unconfiguring 1053
load sharing
algorithms 268
and ESRP don't count 641
and ESRP host attach 641
and VLANs 268, 286
configuring 282
master port 282
maximum ports and groups 279, 280
troubleshooting 286
see also link aggregation
local agent 1773
local database authentication
description 1232
password 1232
user name 1232
local netlogin account
creating 1233
deleting 1236
destination VLAN
modifying 1235
displaying 1236
modifying 1235
local routing database 1089
local-only VRs 2202
locally assigned labels 1064
lockdown timer, MAC 1552
locked entries 1551
log messages 163
log target, EMS
disabling 1751
enabling 1751
logging configuration changes 1765
logging in
stack node 183
logging messages , *see* EMS
logout privilege, network login 1248
loop detection
using ELRP and ESRP 634
using standalone ELRP 2118
loop tests
using ELRP and ESRP 634
using standalone ELRP 2118
loopback interface 430
LPS, tunnel, definition of 1058
LSA type numbers (table)
OSPF 1395, 1416
LSA, description 1395, 1416
LSDB 992
LSDB, description 1395, 1416
LSP
and PW 1069
calculated 1065

LSP *(continued)*
control modes 1062
definition of 1058
introduction 1056
matching next hop 1065
routing 1065
scaling 1105
LSR
definition of 1058
described 1056
egress, definition of 1056
ingress, definition of 1056
LER, description of 1056
locally assigned labels 1064
remotely assigned labels 1064

M

MAC address 158, 172
MAC learning, FDB 832
MAC lockdown
configuring 1551
displaying entries 1551
unconfiguring 1551
MAC lockdown timer
configuring 1555
disabling 1555
displaying entries 1556
displaying the configuration 1555
enabling 1555
examples
active device 1553
inactive device 1553
port movement 1555
overview 1551
understanding 1552
MAC-based
security 832, 1538
VLANs, network login 1260
MAC-based authentication
advantages 1221
configuration, example 1259
configuration, secure MAC 1258
description 1255
disabling 1256
enabling 1256
management access 50
management accounts, displaying 60
Management Information Base , *see* MIBs
management port 69
Management Switch Fabric Module , *see* MSM
manually bind ports 1806
master port, load sharing 282
match conditions
ACL 355
policy 1522
matching
expressions, EMS 1758
LSP next hop 1065
matching *(continued)*
parameters, EMS 1759
maximum bandwidth, QoS 1463
maximum CPU sample limit, sFlow 1775
maximum transmission unit (MTU) 1099
member VLAN 2249
memory protection 134, 144, 145
meters, QoS 1469
mgmt VLAN 69
MIBs, supported 108
minimum bandwidth, QoS 1463
MLD, static 950
modular switch
jumbo frames 263
load sharing, configuring 282
monitor port 308
port number 43, 224
port-mirroring 308
monitor port, port-mirroring 308
monitoring command prompt 51, 53
monitoring the switch 1727
MPLS
configuration example (figure) 1124
definition of 1058
introduction 1056
label stack (figure) 1063
protocol filter 2220, 2221
resetting configuration parameters 1116
sample network (figure) 1056
shim header 1063
shim layer 1062
terms and acronyms 1058
unicast frame on Ethernet 1063
MSDP
anycast RP 1153
default peers 1150
description 1148
mesh-groups 1152
MIBs 1156
peer authentication 1150
peers 1149
PIM border configuration 1149
platforms supported 494, 749, 1149, 1492
policy filter 1151
SA cache 1155
SA cache entry limit 1156
SA request processing 1151
MSM
console sessions 67
MSTI
configuring 1834
enabling 1835
identifier 1834
regional root bridge 1834
root port 1835
see also MSTP
MSTI ID 1834
MSTP

MSTP *(continued)*
advantages of 1830
boundary ports 1835
common and internal spanning tree 1832
configuring 1837
edge safeguard 1820
enabling 1837
hop count 1837
identifiers 1805
inheriting ports 1808
link types
auto 1818
broadcast 1818
configuring 1819
description 1818
edge 1818
point-to-point 1818
multiple spanning tree instances 1833
operation 1838
overview 1829
port roles
alternate 1817
backup 1817
designated 1817
master 1836
port roles
disabled 1817
root 1817
region
configuring 1831, 1832
description 1830
identifiers 1831
see also RSTP
multicast
FDB static entry 830
IGMP 1184
IGMP snooping 1184
IGMP snooping filters 1186
PIM 1163
PIM edge mode 1163
PIM-DM 1164
PIM-SM 1166
PIM-SSM 1167
translation VLAN 2250
Multicast VLAN Registration., *see* MVR
multinetting, *see* IP multinetting
multiple next-hop support 407
multiple routes
IPv6 889, 967
Multiple Spanning Tree Instances , *see* MSTI
Multiple Spanning Tree Protocol , *see* MSTP
multiple supplicants, network login support 1222
MVR
and EAPS 1206–1208
and STP 1208
dynamic 1201
forwarding rules 1204
in a vMAN environment 1210
MVR *(continued)*
static 1201

N

names
character types 34
conventions 34
maximum length of 34
switch 53
VLAN 2222
VLAN, STP, EAPS 34
NAP
and 802.1X 1242
and ACLs 1246
overview 1242
sample scenarios 1243
VSA definitions 1245
native stacking 156
native stacking ports 147
native VLAN, PVST+ 1817
neighbor discovery protocol, LDP 1060
NetBIOS protocol filter 2220, 2221
netlogin
dynamic VLANs
displaying 1265
enabling 1264
example 1265
uplink ports 1264
port restart
description 1266
disabling 1266
displaying 1267
enabling 1266
guidelines 1266
see also network login
Network Access Protection , *see* NAP
network login
authentication methods 1220
campus mode 1225
configuration examples
802.1X 1238
MAC-based 1259
web-based 1252
disabling 1229
disabling, port 1230
enabling 1229
exclusions and limitations 1231
guest VLAN 1239
hitless failover support 1226
ISP mode 1225
local account
deleting 1236
displaying 1236
modifying 1235
local account, destination VLAN
creating 1233
modifying 1235
local database authentication 1232

network login *(continued)*
logout privilege 1248
MAC-based VLANs 1260
move fail action 1230
multiple supplicants 1222
port, enabling 1230
RADIUS attributes 1616, 1618
redirect page 1247
secure MAC 1257
session refresh 1248
settings, displaying 1230
web-based authentication, user login 1253
network VLAN
description 2233
extension to non-PVLAN switch 2235
Next Hop Label Forward Entry (NHLFE), definition of 1058
noAuthnoPriv 120
node
adding to stack 199
removing from stack 210
replacing in stack 202, 204
node address 158
node election
determining primary 85
overview 85
node role
definition 157
election 158
node states 89
node status, viewing 89
non-aging entries, FDB 828
non-isolated subscriber VLAN 2234
non-permanent dynamic entry, FDB 828
non-persistent capable commands 2145
normal area
OSPFv3 1399, 1420
Not-So-Stubby-Area *, see* NSSA
notification tags, SNMPv3 123
notification, SNMPv3 123
NSSA 1399, 1420
see also OSPF

O

opaque LSAs, OSPF 1396
Open LDAP 1638
Open Shortest Path First*, see* OSPF
Open Shortest Path First IPv6*, see* OSPFv3
operational node 158
operators, CLI scripting 2190
ordered LSP control 1062
OSPF
advantages 1508
and ESRP 622
and IP multinetting 918
authentication 1404
configuration example 1406, 1408
consistency 1395
OSPF *(continued)*
database overflow 1395
description 1391, 1508
display filtering 1409
enabling 907
graceful restart 1396, 1417
link type 1401
LSA 1395, 1416
LSDB 1395, 1416
NSSA 1399, 1420
opaque LSAs 1396
point-to-point links 1401, 1422
redistributing routes
configuring 999, 1403, 1424, 1511, 1517
description 434, 998, 1401, 1423, 1510, 1517
enabling or disabling 999, 1403, 1404, 1424, 1511, 1518
restart 1396, 1417
settings, displaying 1409
stub area 1399, 1420
timers 1404
virtual link 1400, 1420
wait interval, configuring 1405
OSPF (Open Shortest Path First) 1149
OSPFv3
advantages 1515
area 0 1398, 1419
areas 1398, 1419
authentication 1425
backbone area 1398, 1419
description 1413, 1515
normal area 1399, 1420
redistributing routes
enabling or disabling 1424
router types 1398, 1419
timers 1425
out-of-profile traffic 1462

P

packet counter
ACL 369
partial sequence number PDU 992
partition 1706
passive interfaces 1163
password security
configuring 57
displaying 60
passwords
creating 57
default 57
displaying 60
failsafe account 55
forgetting 57
local database authentication 1232
security 56
shared secret, TACACS+ 1593, 1595, 1612, 1613
troubleshooting 57
path

path *(continued)*
error message, RSVP 1092
message, RSVP 1091
MTU discovery 265
PBS 1463
peak burst size 1463
peer groups 431
Penultimate Hop Popping, *see* PHP
Per VLAN Spanning Tree, *see* PVST+
permit-established 386
PHP
configuring 1111
definition of 1058, 1064
implicit NULL labels 1064
PIM
and IP multinetting 920
Dense Mode, *see* PIM-DM
mode interoperation 1167
multicast border router (PMBR) 1167
snooping, example 1195–1197
Source Specific Multicast, *see* PIM-SSM
Sparse Mode, *see* PIM-SM
PIM-DM
description 1164
example 1192
PIM-SM
and MSDP 1148
description 1166
example 1193
rendezvous point 1166
PIM-SSM 1167
ping
CCM 1288
troubleshooting 61
PoE
capacitance measurement 1441
configuring 1437
deny port 1439
denying power 1435
disconnect precedence 1435, 1439
EMS message 1437
features 1430
legacy powered devices 1441
operator limit 1441
port labels 1442
port power limits 1437
port priority 1439
resetting ports 1442
SNMP events 1440
upper port power limit 1441
usage threshold 1440
PoE features 1430
poison reverse, RIP 1509, 1516
policies
action statements 1525
autonomous system expressions 1523
examples
policies *(continued)*
examples *(continued)*
translating a route map 1529
translating an access profile 1527
file syntax 1521
rule entry 1521
Policy Based Routing 354
policy file
copying 137
deleting 139
displaying 137
renaming 136
policy match conditions 1522
policy-based routing 402
polling interval, sFlow 1774
port
auto-negotiation 225, 227–229
configuring 223
configuring medium 225, 227–229
cut-through switching 245
duplex setting 225, 227–229
enabling and disabling 224
flow control 225, 227–229, 232
health check link aggregation 278
LFS 238
link aggregation 266
lists 42, 223
load sharing 266
management 69
mode, STP 1846
monitoring display keys 1732
network login 1214
numbers and ranges 42, 223
pause frames 232
priority, STP 1846
receive errors 1729
restart, ESRP 639
restart, netlogin 1266
SNMP trap 224
software-controlled redundant
configuring 327
description 325
speed
configuring 225, 227–229
displaying 331
states, ELSM 1738
supported types of 225, 227–229
transmit errors 1728
utilization 331
viewing
configuration 331
information 331
receive errors 1729
statistics 1728
transmit errors 1728
weight, ESRP 627
wildcard combinations 43, 224
port-based

port-based *(continued)*
load sharing 268
traffic groups 1461
VLANs 2215, 2217
port-mirroring
and ELSM 310
and load sharing 309
and protocol analyzers 308
description 308
displaying 312
examples 311
guidelines 310
monitor port 308
traffic filter 309
ports
alternate stacking 150
native stacking 147
post-authentication VLAN movement, network login 1242
power management
displaying information 98
primary image 1706
primary node
assigning new 216
configuring 169
definition 157
redundancy 147
prioritizing entries, FDB 832
priority
for node role election 158
private AS numbers 434
private community, SNMP 109
privilege levels
admin 51
user 51
privileges
creating 54
default 54
viewing 54
probe, RMON 1777
probeCapabilities 1779
probeDateTime 1779
probeHardwareRev 1779
probeResetControl 1779
probeSoftwareRev 1779
process
control 134
displaying information 141
error reporting 2128
management 140
restarting 141
starting 142
stopping 141
terminating 141
product announcements 8
profile
configuration 2154
description 2135
profile *(continued)*
device detect operation 2142
multiple profiles on a port 2145
obtaining 2144
rules 2144
user authentication operation 2142
prompt
admin account 51, 53
unsaved changes 53
user account 51, 53
propagating labels 1060
protected VLAN
EAPS 562
STP 1802
protocol analyzers, use with port-mirroring 308
protocol filters 2220, 2221
Protocol Independent Multicast. *, see* PIM
protocol-based VLANs 2220, 2221
proxy ARP
communicating with devices outside subnet 916
conditions 914
configuring 914
description 914
MAC address in response 914
responding to requests 914
subnets 916
pseudonode 994
PSNP 992
psuedo wire *, see* PW
public community, SNMP 109
PVLAN
components 2233
configuration example 2243
FDB entries 2236
Layer 3 communications 2238
limitations 2238, 2250
MAC address management 2236
over mulitple switches 2234
VLAN translation component 2230
PVST+
description 1804, 1816
native VLAN 1817
VLAN mapping 1816
PW, and LSPs 1069

Q

Q-in-Q 2261
QoS
802.1p replacement 1477
and RSVP 1089
applications and guidelines 1455
committed information rate 1463
database applications 1456
DiffServ model 1105
displaying mapping information 1120
DSCP replacement 1478
dual-rate 1463
egress port 1472

QoS *(continued)*
egress profiles
default configuration 1470
EXP bits 1105
explicit packet marking 1457
maximum bandwidth 1463
metering 1469
meters 1469
minimum bandwidth 1463
peak burst size 1463
profiles
default DSCP mapping 1459
rate specification 1463
scheduling 1464
single-rate 1462
strict priority queuing 1464
three-color 1463
traffic groups
802.1p-based 1458
ACL-based 1458
CoS-based 1458
DiffServ-based 1459
introduction 1457
port-based 1461
precedence 1462
VLAN-based 1461
troubleshooting 1469, 1480, 1481
two-color 1462
use with full-duplex links 1456
viewing port settings 331
VLANs
flood control 1487
voice applications 1456
web browsing applications 1456
weighted round robin 1464
Quality of Service (QoS) 161

R

RADIUS
and TACACS+ 71, 1232, 1591, 1593, 1597, 1612
description 71
schema modification 1635
TCP port 1611
use with Universal Port 1638, 2153
RADIUS accounting
disabling 1614
enabling 1614
RADIUS attributes, network login 1616, 1618
RADIUS client
configuring 1611
rapid root failover 1808
Rapid Spanning Tree Protocol *, see* RSTP
rate limiting
introduction 1462
rate shaping
introduction 1462
rate specification
QoS 1463
rate-limiting
disabling 1464
rate-shaping
disabling 1464
read-only switch access 109
read-write switch access 109
reboot
switch 1708
receive errors, port 1729
redirect page, network login 1247
redirection, URL *, see* URL redirection
redundancy
in a stack 147
redundant ports, software-controlled
configuring 327
description 325
refresh, ACLs 1451
regions, MSTP 1830
relative route priorities
IPv6 890, 968
Remote Authentication Dial In User Service*, see* RADIUS
remote collector 1773
Remote Monitoring*, see* RMON
remotely assigned label 1064
renaming a VLAN 2226
reservation attributes and styles (table) 1092
reservation error message 1092
reservation message 1091
reservation requests 1090
reservation styles 1092
resilience 550
responding to ARP requests 914
restart process 141
restart, graceful 1396, 1417
RFC 1112 1157
RFC 1195 992, 995
RFC 1256 884
RFC 1542 927
RFC 1745 421
RFC 1771 421
RFC 1812 884
RFC 1965 421
RFC 1966 421
RFC 1997 421
RFC 2113 1091
RFC 2236 1157
RFC 2338 921, 2276
RFC 2385 421, 1150
RFC 2439 421
RFC 2460 955
RFC 2461 959
RFC 2462 964
RFC 2545 421
RFC 2576 2060
RFC 2613 2071
RFC 2787 921, 2276
RFC 2796 421

RFC 2918 421
RFC 2961 1104
RFC 3032 1063
RFC 3046 924
RFC 3107 421
RFC 3209 1090, 1094, 1096, 1104
RFC 3376 1184
RFC 3392 421
RFC 3410 2060
RFC 3411 2060
RFC 3412 2060
RFC 3413 2060
RFC 3414 2060
RFC 3415 2060
RFC 3418 1046
RFC 3446 494, 749, 1149, 1492
RFC 3513 957
RFC 3618 494, 749, 1148, 1149, 1492
RFC 3619 550
RFC 3787 997
RFC 3826 2060
RFC 4090 1102
RFC 4271 421
RFC 4291 955
RFC 4360 421
RFC 4447 1083
RFC 4456 421
RFC 4486 421
RFC 4760 421
RFC 4893 421
RFC 5085 1146
RFC 5396 421
RFC-4090 1102
RFCs
    BGP 421
    bridge 1846
    IPv4 multicast routing 1157
    IPv4 unicast routing 884
    IPv6 unicast routing 955, 964
    RIPng 1514
    VRRP 921, 2276
ring topology 149
RIP
    advantages 1508
    and IP multinetting 919
    configuration example 1512
    description 1508
    disabling route advertising 1510, 1517
    enabling 907
    limitations 1508
    poison reverse 1509, 1516
    redistributing routes
        configuring 999, 1403, 1424, 1511, 1517
        description 434, 998, 1401, 1423, 1510, 1517
        enabling or disabling 1511
    split horizon 1509, 1516
    version 2 1510
RIP (Routing Information Protocol) 1149
RIPng
    advantages 1515
    configuration example 1518, 1519
    description 1515
    limitations 1515
    redistributing routes
        enabling or disabling 1518
    routing table entries 1509, 1516
    triggered updates 1509, 1516
RMON
    agent 1777
    alarm actions 1781
    configuring 1780
    description 1777
    Events group 1779
    features supported 1778
    History group 1779
    management workstation 1777
    output 1781
    probe 1777
    probeCapabilities 1779
    probeDateTime 1779
    probeHardwareRev 1779
    probeResetControl 1779
    probeSoftwareRev 1779
    trapDestTable 1780
Rolling Upgrade 188–190
route aggregation 451
route confederations 429
Route Distinguisher 1026
route leaking 997
route recording, RSVP 1098
route reflectors 427
route selection 433
router interfaces 885, 956
router types
    OSPFv3 1398, 1419
Routing Information Protocol, *see* RIP
Routing Information Protocol, IPv6, *see* RIPng
routing protocols
    adding to virtual routers 2208
routing table
    entries, RIPng 1509, 1516
    IPV4, populating 888
    IPv6, populating 965
RP
    and MSDP 1148
    definition 1166
RSTP
    and STP 1829
    configuring 1845
    designated port rapid behavior 1824
    edge safeguard 1820
    link types
        auto 1818
        broadcast 1818
        configuring 1819
        description 1818

RSTP *(continued)*
link types *(continued)*
edge 1818
point-to-point 1818
operation 1822
overview 1817
port roles
alternate 1817
backup 1817
designated 1817
disabled 1817
edge 1820
root 1817
rapid reconvergence 1825
receiving bridge behavior 1825
root port rapid behavior 1823
timers 1821
topology information, propagating 1825
see also STP
RSVP
and QoS 1089
definition of 1058, 1089
explicit route 1095, 1096
fixed filter reservation style 1092
label 1095
label request 1095
LSP scaling 1105
message types 1090
objects 1094
path error message 1092
path message 1091
record route 1095
reservation error message 1092
reservation message 1091
reservation requests 1090
reservation styles 1092
route recording 1098
session attribute 1095
shared explicit reservation style 1092
traffic engineering overview 1090
tunneling 1094
wildcard reservation style 1092
RSVP-TE
extended tunnel ID 1094
multiple RSVP-TE LSPs 1103
overview 1090
secondary RSVP-TE LSPs
1101
RSVP-TE, definition of 1058
rule entry
ACL 346
policy 1521
rule syntax, ACL 346
rule types 504

S

s-tag ethertype translation 2268
Samba
Samba *(continued)*
schema 1640
use with LDAP 1640
use with RADIUS-to-LDAP mappings 1634
sampling rate, sFlow 1775
saving configuration changes 1712
scheduling, QoS 1464
scoped IPv6 addresses 958
SCP2 1664
secondary image 1706
secure MAC
configuration, example 1258
description 1257
Secure Shell 2 , *see* SSH2 protocol
Secure Socket Layer, *see* SSL
security
egress flooding 833
security name, SNMPv3 113
service provide 1058
service tag 2262
session refresh, network login 1248
sessions
console 67
deleting 78
maximum number of 67
shell 68
SSH2 81, 1659
Telnet 73
TFTP 83
sFlow
configuration example 1776
configuring 1772
displaying configuration 1777
displaying statistics 1777
enabling
on specific ports 1774
on the switch 1774
local agent 1773
maximum CPU sample limit 1775
polling interval 1774
remote collector 1773
resetting values 1776
sampling rate 1775
shared explicit reservation style 1092
shared secret
TACACS+ 1593, 1595, 1612, 1613
shell
configuring 68
maximum number of 68
overview 68
shim header
described 1062
illustration 1062
shim layer 1062
show eaps counters 571, 572
Simple Network Management Protocol , *see* SNMP
Simple Network Time Protocol, *see* SNTP
single-rate QoS 1462

slapd 1639
slot
  diagnostics 1734
slot number 147, 168
slow path routing 896
Smart Redundancy
  configuring 327
  description 325
  displaying 331
  port recovery 325
smart refresh, ACLs 1451
SMON 1781, 2071
SNAP protocol 2222
SNMP
  community strings 109
  configuring 108
  settings, displaying 110
  supported MIBs 108
  system contact 109
  system location 110
  system name 110
  trap receivers 108
  using 106
SNMPEngineBoots 112
snmpEngineID 112
SNMPEngineTime 112
SNMPv3
  groups 119
  MIB access control 121
  notification 123
  overview 110
  security 112
  security name 113
  tags, notification 123
  target address 123
  user name 113
SNTP
  configuring 126
  Daylight Savings Time 126
  description 126
  example 130
  Greenwich Mean Time offset 126
  Greenwich Mean Time Offsets (table) 129
  NTP servers 126
software image , *see* image
software module
  .xmod file 1711
  activating 1711
  description 1711
  downloading 1707
  uninstalling 1711
software requirements for switches 11
software signature 1710
software-controlled redundant ports
  and link aggregation 267
  description 325
  displaying 331
  displaying configuration 327
software-controlled redundant ports *(continued)*
  troubleshooting 325
  typical configurations 325
source active (SA) message 1148
source IP lockdown
  clearing information 1576
  configuring 1575
  displaying information 1576
  overview 1574
spanning tree identifier, *see* StpdID
Spanning Tree Protocol , *see* STP
speed, displaying setting 331
speed, ports
  configuring 225, 227–229
  displaying 331
split horizon, RIP 1509, 1516
SSH2 client 1663
SSH2 protocol
  ACL policy 1660
  authentication key 1648
  default port 1658
  description 81
  maximum number of sessions 81, 1659
  sample ACL policies 1660
  TCP port number 1658
SSL
  certificates, downloading 1677
  certificates, generating 1677
  certificates, pregenerated 1678
  description 1676
  disabling 1677
  displaying information 1680
  enabling 1676
  private key, downloading 1678
  private key, pregenerated 1679
  secure web access 1676
stack 156
  see also SummitStack
stack number indicator 147
stackable switch 156
stacking
  adding a node 199
  available methods 152
  backup 147
  configuring 164, 168
  control plane redundancy 159
  daisy chain 149, 213
  definition 147
  dismantling stack 211
  dual master condition 149
  dual master situation 214, 215
  dual primary situation 213
  gateway 192
  hitless failover 159
  IP address 192
  LEDs 147
  license mismatch 220
  link failure 221

stacking *(continued)*
log messages 163
MAC address 172
merging 205
native stacking ports 147
no primary node 216
no primary-capable node 216
ports, native and alternate 150
primary 147
priority 147
Quality of Service (QoS) 161
redundancy 147
removing a node 210
replacing a node 202, 204
ring topology 149
roles 159, 169
slot number 147, 168
stack number indicator 147
terminology 156
traps 221
troubleshooting 147, 211–213, 216, 218–221
verifying configuration 175–178
stacking link 156
stacking port 156
stand-alone switch
load sharing example 285
standby node
configuring 169
definition 157
start process 142
startup screen
switch 53
static IGMP 1187
static MLD 950
static MVR 1201
static networks, and BGP 435
static routes 888, 966
statistics
CPU utilization 1782
port 1728
status monitoring 1727
sticky threshold, ELSM 1744
stop process 141
STP
advanced example 1813
and ESRP 633
and IP multinetting 920
and MVR 1208
and RSTP 1829
and VLANs 1801
and VRRP 2278
autobind ports 1807
basic configuration example 1810
bridge priority 1846
compatibility between 802.1D-1998 and 802.1D-2004 1788
configurable parameters 1846
configuring 1845
STP *(continued)*
description 1787
displaying settings 331, 1847
domains
802.1D 1802
802.1w 1802
creating 1801
deleting 1801
description 1801
displaying 1847
mstp 1803
EMISTP
example 1812
rules 1814
encapsulation mode
802.1D 1804
description 1804
EMISTP 1804
PVST+ 1804
forward delay 1846
guidelines 1844
hello time 1846
hitless failover support 1809
inheriting ports 1808
manually bind ports 1806
max age 1846
max hop count 1846
MSTI ID 1846
names 34
path cost 1846
port and multiple STPDs 1801
port mode 1846
port priority 1846
port states
blocking 1805
disabled 1806
displaying 1847
forwarding 1806
learning 1806
listening 1805
protected VLAN 1802
PVST+, description 1816
rapid root failover 1808
rules and restrictions 1844
StpdID 1805, 1846
troubleshooting 1844
STPD 2116
StpdID 1805
strict priority queuing 1464
strings, community 109
stub area, OSPF 1399, 1420
subcomponents, EMS 1755
Subnetwork Access Protocol *, see* SNAP protocol
subscriber VLAN
description 2233
extension to non-PVLAN switch 2235
subVLAN 931
SummitStack

SummitStack *(continued)*
available methods 152
logging in 183
managing licenses 184
path 156
segment 159
state 159
topology 148, 156
troubleshooting 211–213, 218–221
SummitStack configuration 146
SummitStack-V feature 150, 152
superVLAN 931
supplicant
collecting information 2150
configuration parameters 2151
description 2150
Windows XP client configuration 1643
supplicant side requirements 1237
support
technical support 8
switch
adding to stack 199
reboot 1708
replacing in stack 202, 204
software requirement 11
startup screen 53
switch management
console 69
overview 67
TFTP 83, 84
user sessions 67
switch name 53
switch RMON features 1778
switch series, table 10
symbols, command syntax 42
syntax
abbreviated 37
understanding 32
see also CLI
syntax helper 33
system contact, SNMP 109
system diagnostics 1734
system health check
monitoring 1736
system health checker
description 1737
displaying 1737
mode of operation 1737
system LEDs 2108
system location, SNMP 110
system name, SNMP 110
system odometer 2131
system redundancy
bulk checkpointing 88
determining the primary node 85
dynamic checkpointing 88
failover 86
node election 85
system redundancy *(continued)*
relaying configurations 87
viewing
checkpoint statistics 88
status 89
system temperature 1749
system virtual routers 2199

T

TACACS+
and RADIUS 71, 1232, 1591, 1593, 1597, 1612
configuration example 1594
configuring 1592
disabling 1593, 1612
enabling 1593, 1612
password 1593, 1595, 1612, 1613
TACACS+ accounting
disabling 1596
enabling 1596
tagged VLAN (802.1Q) 2217
target address, SNMPv3 123
TCL functions 2192
TCP MD5 authentication 1150
TCP port number 77
technical support
contacting 8
Telnet
ACL policy 78
and safe defaults mode 77
changing port 76
client 72
configuring virtual router 76
connecting to another host 73
controlling access 76
default port 73
default virtual router 73
description 72
disabling 77
displaying status 77
re-enabling 77
sample ACL policies 80
server 73
session
establishing 72
maximum number of 73
opening 72
terminating 77
viewing 77
TCP port number 73
using 72
temperature
displaying 1750
Terminal Access Controller Access Control System Plus*, see* TACACS+
terminate process 141
TFTP
connecting to another host 84
default port 84

TFTP *(continued)*
description 83
maximum number of sessions 83
server 1704
server requirements 83
using 83, 1719
TFTP server, troubleshooting 83
three-color Qos 1463
time trigger 2141
timeout interval, EDP 319
timeout, MAC lockdown 1551
TLS
802.1Q encapsulation 1070
basic configuration example (figure) 1131
characteristics 1074
toggling, ESRP modes of operation 627
Tool Command Language , *see* TCL
TOP command 2130
TOS 1459
traceroute
CCM 1288
tracking
example 639
traffic engineering (TE), definition of 1058
traffic filter, port-mirroring 309
traffic groups
ACL-based 1458
DiffServ-based 1459
port-based 1461
precedence 1462
VLAN-based
802.1p-based 1458
traffic groups, introduction 1457
traffic, in-profile 1462
traffic, out-of-profile 1462
translation VLAN 2249
see also PVLAN
transmit errors, port 1728
trap receivers, SNMP 108
trapDestTable 1780
traps
generated by stacks 221
trigger
configuration 2154
device 2138
EMS event 2141
time 2141
user authentication 2139, 2140
triggered updates, RIPng 1509, 1516
Trivial File Transfer Protocol., *see* TFTP
troubleshooting
ACLs 1449
ASCII-formatted configuration file 1716
campus mode 1225
connectivity 61
CPU utilization 1782
debug mode, EMS 2127
diagnostics
troubleshooting *(continued)*
diagnostics *(continued)*
viewing results 1736
downloads and TFTP 83
EAPS
loop protection messages 567
ring ports 563
ESRP 629, 632
filenames 135, 136, 2130
guest VLAN configuration 1240
IP fragmentation 265
ISP mode 1225
Layer 1 2107
LEDs 1736
link aggregation 266
LLDP 1041
load sharing 266, 279, 280, 282
memory 144, 145
passwords 57
path MTU discovery 265
PoE 1435
port configuration 2114
port-mirroring
guidelines 309
QoS 1480
required software 2115
software limits 2115
SSL 54
stacks 211–213
STP 1844
system LEDs 2108
TFTP server 83
VLANs 2227
vMANs 264
VRRP and ESRP 2279
troubleshooting MPLS 1056
troubleshooting stack connections 147
trunks 2218
Tspec object 1090, 1091
tunneling
IP 956, 985, 987
two-color Qos 1462
Type-of-Service 1459

U

UDP echo server 929
unicast traffic, translation VLAN 2250
Universal Port
configuration 2154
configuration overview 2151
dynamic profiles 2137
Handset Provisioning Module
obtaining 2144
sample profiles 2160, 2163
non-persistent capable commands 2145
overview 2135
profile 2135
static profiles 2136

Universal Port *(continued)*
supported commands 2145
troubleshooting 2158
use with central directory service 2153
use with Open LDAP 1638
variables 2147
upgrading the image 1704
uplink ports, netlogin 1264
uploading
ASCII-formatted configuration 1716
XML-formatted configuration 1719
upstream forwarding 833
URL redirection 1221
user account 51, 54
user authentication trigger 2139, 2140
user name, local database authentication 1232
user name, SNMPv3 113
user sessions 67
see also sessions
user virtual routers 2200
User-Based Security Model , *see* USM
users
access levels 50
adding 54
authenticating 71, 1232
creating 54
default 54
deleting 54
passwords 57
viewing 54
USM, SNMPv3 security 112
utilization, port 331

V

variables, CLI scripting 2187
vendor ID 1245, 1619–1623, 1625, 1626
Vendor Specific Attribute, *see* VSA
version string 1709
video applications 1456
video applications, and QoS 1456
View-Based Access Control Model, SNMPv3 121
viewing information 331
virtual link
OSPF 1400, 1420
virtual private LAN (VPN), definition of 1058
virtual router , *see* VR
virtual router (VR) 2118
Virtual Router Redundancy Protocol, *see* VRRP
Virtual Router Redundancy Protocol. , *see* VRRP
virtual routers
commands 2360
default for Telnet 73
VLAN 98, 2115, 2116
VLAN aggregation
description 931
limitations 932
properties 932
proxy ARP 933
VLAN aggregation *(continued)*
secondary IP address 931
superVLAN 931
VLAN isolation 2231
VLAN stacking 2261
VLAN tagging 2217
VLAN translation
broadcast traffic behavior 2250
component of PVLAN 2230
configuration 2251
description 2249
member VLAN 2249
multicast traffic behavior 2250
translation VLAN 2249
unicast traffic behavior 2250
VLAN-based traffic groups 1461
VLAN, guest, *see* guest VLAN
VLANid 2217
VLANs
and load sharing 268, 286
and STP 1801
and virtual routers 2214
assigning a tag 2218
benefits 2213
configuration examples 2227
default tag 2218
default VLAN 2222
description 2213
disabling 2226
disabling route advertising 1510, 1517
displaying settings 331
enabling 2226
IPv4 routing 907
IPv6 address 2227
IPv6 addresses 2213
IPv6 routing 970
mgmt 69
mixing port-based and tagged 2220
names 34, 2222
port-based 2215, 2217
protocol filters
customizing 2221
deleting 2222
predefined 2220, 2221
protocol-based 2220, 2221
renaming 2226
tagged 2217
troubleshooting 2227
trunks 2218
types 2214
untagged packets 2218
VLANid 2217
vMANs
and MVR 1210
configuring 2271
jumbo frames 264
names 34
s-tag ethertype translation 2268

vMANs *(continued)*
troubleshooting 2271
voice applications, and QoS 1456
VPLS
definition of 1058
domains, configuring 1127
VR
adding routing protocols 2208
and VLANs 2214
commands 2205
configuration domain 2205
configuration example 2212
creating 2206, 2207
deleting 2206
description 2198
displaying information 2210
system 2199
user 2200
VR-Control 2199
VR-Default 2199
VR-Mgmt 2199
VR-Control virtual router 2199
VR-Default virtual router 2199
VR-Mgmt virtual router 2199
VRRP
advertisement interval 2280
and ESRP 633, 2279
and STP 2278
default gateway 618, 2276
description 618, 2276
electing the master 2280
examples 2303
hitless failover support 2288
master down interval 2280
master router
determining 2279
electing 2279
preemption 2281
multicast address 2278
ping tracking 2282
priority 2280
redundancy 2302
route table tracking 2282
skew time 2280
tracking
description 2282
virtual router MAC address 2282
VLAN tracking 2282
VRs 2202
VSA
203
example 1619–1623, 1625, 1626
204
example 1619–1623, 1625, 1626
205
example 1619–1623, 1625, 1626
206
examples 1619–1623, 1625, 1626
VSA *(continued)*
definitions
Extreme 1619–1623, 1625, 1626
NAP 1245
order of use 1619–1623, 1625, 1626

W

web browsing applications, and QoS 1456
web-based authentication
advantages 1221
configuration, example 1252
disabling 1246
disadvantages 1221
enabling 1246
requirements 1220
URL redirection 1221
user login setup 1253
weighted deficit round-robin priority
QoS
weighted deficit round-robin priority 1464
weighted round robin queuing 1464
wildcard combinations, port 43, 224
wildcard reservation style 1092

X

XML configuration mode 139
XML-formatted configuration file 139