CONFIGURATION GUIDE

Extreme SLX-OS Layer 2 Switching Configuration Guide, 18s.1.01a

Supporting the ExtremeSwitching SLX 9140 and SLX 9240 Switches

9036038-00
February 2019

Copyright © 2019 Extreme Networks, Inc. All Rights Reserved.

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, please see: www.extremenetworks.com/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: www.extremenetworks.com/support/policies/software-licensing

Contents

Preface **9**
- Conventions 9
  - Notes, cautions, and warnings 9
  - Text formatting conventions 9
  - Command syntax conventions 10
- Documentation and Training 10
  - Training 10
- Getting Help 10
  - Subscribing to Service Notifications 11
- Providing Feedback to Us 11

About This Document **13**
- Supported hardware and software 13
- What's new in this document 13

Link Aggregation **15**
- Link aggregation overview 15
  - Link Aggregation Control Protocol 16
  - LAG distribution process and conditions 16
- Configuring and managing Link Aggregation 16
  - Configuring a new port channel interface 17
  - Deleting a port channel interface 17
  - Adding a member port to a port channel 17
  - Deleting a member port from a port channel 18
  - Configuring the minimum number of LAG member links 18
  - Configuring the LACP system priority 19
  - Configuring the LACP port priority 19
  - Configuring the LACP timeout period 20
  - Configuring LACP default Up 20
  - Troubleshooting LACP 21
- LAG load balancing 21
- Show and clear LAG commands 22
  - Displaying port-channel information 22
  - Displaying LAG hashing 23
  - Displaying LACP system-id information 24
  - Displaying LACP statistics 24
  - Clearing LACP counter statistics on a LAG 24
  - Clearing LACP counter statistics on all LAG groups 24

VLANs **25**
- 802.1Q VLAN overview 25
- Configuring VLANs 25
  - Configuring a VLAN 25
  - Configuring a switchport interface 25
  - Configuring the switchport interface mode 26
  - Configuring the switchport access VLAN type 26
  - Configuring a VLAN in trunk mode 27
  - Configuring a native VLAN on a trunk port 27

Enabling VLAN tagging for native traffic....28
Displaying the status of a switchport interface....29
Displaying the switchport interface type....29
Verifying a switchport interface running configuration....30
Displaying VLAN information....30
VLAN statistics....30
Enabling statistics on a VLAN....31
Displaying statistics for VLANs....31
Clearing statistics on VLANs....32
Configuring the MAC address table and conversational MAC learning....32
Conversational MAC learning....32
Specifying or disabling the aging time for MAC addresses....32
Adding static addresses to the MAC address table....33
Enabling conversational MAC learning (CML)....33
Configuring virtual routing interfaces....34
DHCP Layer 2 relay agent Option 82 for VLANs....34
Overview....34
Option 82....34
Topologies and operation....35
Considerations for Layer 2 relay agent....38
Configuring DHCP Layer 2 relay agent Option 82....38
Endpoint tracking....39
Overview....39
Additional details and considerations....39
RADIUS VSA support....40

VXLAN Extension Tunnels....43
VXLAN extension tunnels overview....43
Layer 2 data path....44
Layer 3 data path....45
Logical VTEP control plane....45
VXLAN resiliency....45
Configuring VXLAN extension tunnels....46
Configuring logical VTEPs....47

IP-Based Management Cluster....49
IP-based management cluster....49
Configuring an IP-based management cluster....50

Overlay Services....51
Overlay services overview....51
Overlay services provisioning....52
Overlay policy map rules....53
Overlay policy map behavior....53
Overlay class map rules....53
Flow-based classification rules....54
Flow-based classification behavior....56
Overlay policy system maximum numbers....57
Overlay policy-map inner flow keys and actions supported....58
ACL matches supported....58
IPv4 ACL keys supported....58
Flow-based QoS....58

Security ACLs....59
Overlay services interaction with hardware profiles....59
Overlay visibility profiles....59
Overlay gateway bindings and TCAM profiles....59
Using overlay policy map commands....60
Overlay configuration examples....61

Logical Interfaces....63
Logical interfaces overview....63
LIFs and bridge domains....63
Configuration considerations....63
Configuring a logical interface on a physical port or port-channel (LAG)....65

Bridge Domains....67
Bridge domain overview....67
Bridge domain statistics....67
Configuring a bridge domain....67
Displaying bridge-domain configuration information....68
Enabling statistics on a bridge domain....71
Displaying statistics for logical interfaces in bridge domains....72
Clearing statistics on bridge domains....73

Multi-Chassis Trunking (MCT)....75
MCT Overview....75
MCT terminology....76
MCT peer link....77
Cluster control VLAN....77
SLX-OS MCT control plane....77
MCT data plane traffic forwarding....80
Configuration considerations....82
Configuring the BGP EVPN peer....83
Configuring the MCT domain between a leaf switch pair....84
Configuring the Leaf1 cluster and client....85
Configuring the Leaf2 cluster and client....86
Configuring additional MCT cluster parameters....88
Changing the client-isolation mode....88
Changing the designated-forwarder hold timer value....88
Enabling DF load balancing....88
Changing the cluster control VLAN....88
Moving the traffic from an MCT node to the remote node....89
Configuring an auto-generated ESI for a cluster client....89
Displaying MCT information....89
Displaying the cluster information....89
Displaying the cluster client information....90
Displaying and clearing the MAC address table cluster information....90
Loop prevention in MCT through STP....91
Bridge ID....91
Port ID....92
Loop prevention configuration considerations....92
Configuring the last byte of the bridge ID....93
Displaying the root port on the cluster....93
Bridge domain for Layer 2 multitenancy....94

MCT bridge domain example....94
Configuration considerations for bridge domain for MCT....95
Configuring a bridge domain....95
Enabling and displaying bridge domain statistics....97
BFD support for Layer 3 protocols on MCT....97
BFD packet transmission....98
BFD packet reception....98
Enabling Layer3 routing for an MCT VLAN....98

802.1d Spanning Tree Protocol....101
Spanning Tree Protocol overview....101
Spanning Tree Protocol configuration notes....101
Optional features....101
STP states....102
BPDUs....102
TCN BPDUs....103
STP configuration guidelines and restrictions....103
Understanding the default STP configuration....103
STP features....104
Root guard....104
BPDU guard....105
Error disable recovery....105
PortFast....106
STP parameters....106
Bridge parameters....106
Error disable timeout parameter....107
Port-channel path cost parameter....107
Configuring STP....108
Enabling and configuring STP globally....108
Enabling and configuring STP on an interface....110
Configuring basic STP parameters....112
Re-enabling an error-disabled port automatically....114
Clearing spanning tree counters....115
Clearing spanning tree-detected protocols....115
Shutting down STP....116

802.1w Rapid Spanning Tree Protocol....117
Rapid Spanning Tree Protocol overview....117
RSTP parameters....118
Edge port and automatic edge detection....118
Configuring RSTP....118
Enabling and configuring RSTP globally....118
Enabling and configuring RSTP on an interface....120
Configuring basic RSTP parameters....123
Clearing spanning tree counters....125
Clearing spanning tree-detected protocols....125
Shutting down RSTP....126

Per-VLAN Spanning Tree+ and Rapid Per-VLAN Spanning Tree+....127
PVST+ and R-PVST+ overview....127
PVST+ and R-PVST+ guidelines and restrictions....127
PVST+ and R-PVST+ parameters....128

Bridge protocol data units in different VLANs....128
BPDU configuration notes....128
PortFast....132
Edge port and automatic edge detection....132
Configuring PVST+ and R-PVST+....132
Enabling and configuring PVST+ globally....132
Enabling and configuring PVST+ on an interface....134
Enabling and configuring PVST+ on a system....136
Enabling and configuring R-PVST+ globally....142
Enabling and configuring R-PVST+ on an interface....143
Enabling and configuring R-PVST+ on a system....145
Clearing spanning tree counters....152
Clearing spanning tree-detected protocols....152
Shutting down PVST+ or R-PVST+....153

802.1s Multiple Spanning Tree Protocol....155
MSTP overview....155
Common Spanning Tree (CST)....155
Internal Spanning Tree (IST)....155
Common Internal Spanning Tree (CIST)....155
Multiple Spanning Tree Instance (MSTI)....156
MST regions....156
MSTP regions....156
MSTP guidelines and restrictions....156
Interoperability with PVST+ and R-PVST+....157
MSTP global level parameters....157
MSTP interface level parameters....158
Edge port and automatic edge detection....158
BPDU guard....158
Restricted role....159
Restricted TCN....159
Configuring MSTP....159
Enabling and configuring MSTP globally....160
Enabling and configuring MSTP on an interface....163
Enabling MSTP on a VLAN....165
Configuring basic MSTP parameters....166
Clearing spanning tree counters....169
Clearing spanning tree-detected protocols....169
Shutting down MSTP....169

Link-Fault Signaling (LFS)....171
LFS overview....171
LFS configuration guidelines....171
Configuring LFS globally....171
Configuring LFS at interface level....172
LFS show and clear commands....172

Preface

- Conventions 9
- Documentation and Training 10
- Getting Help 10
- Providing Feedback to Us 11

This section discusses the conventions used in this guide, ways to provide feedback, additional help, and other Extreme Networks® publications.

Conventions

This section discusses the conventions used in this guide.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.

Format	Description
`Courier font`	Identifies CLI output. Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

Current Product Documentation	www.extremenetworks.com/documentation/
Archived Documentation (for earlier versions and legacy products)	www.extremenetworks.com/support/documentation-archives/
Release Notes	www.extremenetworks.com/support/release-notes
Hardware/Software Compatibility Matrices	https://www.extremenetworks.com/support/compatibility-matrices/
White papers, data sheets, case studies, and other product resources	https://www.extremenetworks.com/resources/

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

Getting Help

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: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. 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 and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) 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

Subscribing to Service Notifications

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

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

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

4. Click **Submit**.

Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

Please provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Document

- Supported hardware and software....13
- What's new in this document....13

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Extreme Networks, Inc. for this SLX-OS release, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release:

- ExtremeSwitching SLX 9140
- ExtremeSwitching SLX 9240

NOTE
Some of the commands in this document use a slot/port designation. Because the SLX 9140 and the SLX 9240 do not contain line cards, the slot designation must always be "0" (for example, 0/1 for port 1).

What's new in this document

The following table includes descriptions of new information added to this guide for SLX-OS Release 18s.1.01a.

TABLE 1 Summary of enhancements in SLX-OS Release 18s.1.01a

Feature	Description	Described in
Endpoint tracking	The endpoint tracking feature minimizes the configuration and management of VLANs on switches in the data center.	Endpoint tracking on page 39

Link Aggregation

- Link aggregation overview 15
- Configuring and managing Link Aggregation 16
- LAG load balancing 21
- Show and clear LAG commands 22

Link aggregation overview

Link aggregation enables you to bundle multiple physical Ethernet links into a single logical trunk, providing enhanced performance and redundancy.

The aggregated trunk is referred to as a Link Aggregation Group (LAG) or *port-channel*. The LAG is viewed as a single link by connected devices, the Spanning Tree Protocol, IEEE 802.1Q VLANs, and so on. When one physical link in the LAG fails, the other links stay up. A small drop in traffic is experienced when the link carrying the traffic fails.

To configure links to form a LAG, the physical links must be of the same speed. Link aggregation can be done by statically configuring the LAG, or by dynamically configuring the LAG using the IEEE 802.1AX Link Aggregation Control Protocol (LACP).

When queuing traffic from multiple input sources to the same output port, all input sources are given the same weight, regardless of whether the input source is a single physical link or a trunk with multiple member links.

The benefits of link aggregation are as follows:

- Increased bandwidth (The logical bandwidth can be dynamically changed as the demand changes.)
- Increased availability
- Load sharing
- Rapid configuration and reconfiguration

Each LAG consists of the following components:

- A MAC address that is different from the MAC addresses of the LAG's individual member links.
- An interface index for each link to identify the link to the neighboring devices.
- An administrative key for each link. Only the links with the same administrative key value can be aggregated into a LAG. On each link configured to use LACP, LACP automatically configures an administrative key value equal to the port-channel identification number.

Two LAG types are supported:

- Static LAG— In static link aggregation, links are added into a LAG without exchanging any control packets between the partner systems. The distribution and collection of frames on static links is determined by the operational status and administrative state of the link.
- Dynamic, standards-based LAG using LACP—Dynamic link aggregation uses LACP to negotiate with links that can be added and removed from a LAG. Typically, two partner systems sharing multiple physical Ethernet links can aggregate a number of those physical links using LACP. LACP creates a LAG on both partner systems and identifies the LAG by the LAG ID. All links with the same administrative key, and all links that are connected to the same partner switch become members of the LAG. LACP continuously exchanges LACPDUs to monitor the health of each member link.

The following trunk types are supported:

- Static and standards-based LAG

The following LAG scalability configurations are supported:

- The SLX 9140 supports 72 LAGs with each containing up to 64 ports.
- The SLX 9240 supports 128 LAGs with each containing up to 64 ports.

Link Aggregation Control Protocol

Link Aggregation Control Protocol (LACP) is an IEEE 802.1AX standards-based protocol that allows two partner systems to dynamically negotiate attributes of physical links between them to form logical trunks. LACP determines whether a link can be aggregated into a LAG. If a link can be aggregated into a LAG, LACP puts the link into the LAG. All links in a LAG inherit the same administrative characteristics.

LACP operates in two modes:

- *Active mode*— LACP initiates the LACPDU exchange regardless of whether the partner system sends LACPDUs.
- *Passive mode* — LACP responds to Link Aggregation Control Protocol Data Units (LACPDUs) initiated by its partner system but does not initiate the LACPDU exchange.

LAG distribution process and conditions

The LAG aggregator is associated with the collection and distribution of Ethernet frames. The collection and distribution process implements:

- Inserting and capturing control PDUs.
- Restricting the traffic of a given conversation to a specific link.
- Load balancing between individual links.
- Handling dynamic changes in LAG membership.

On each port, link aggregation control does the following:

- Maintains configuration information to control port aggregation.
- Exchanges configuration information with other devices to form LAGs.
- Attaches ports to and detaches ports from the aggregator when they join or leave a LAG.
- Enables or disables an aggregator's frame collection and distribution functions.

LAG configuration guidelines:

- Each link can be associated with a LAG; a link cannot be associated with more than one LAG. The process of adding and removing links to and from a LAG is controlled statically or dynamically (through LACP).
- Interfaces configured as switchport interfaces cannot be aggregated into a LAG. However, a LAG can be configured as a switchport.

Configuring and managing Link Aggregation

The following sections discuss working with Link Aggregation on Extreme devices.

Configuring a new port channel interface

Follow this procedure to create a new port channel interface at the global configuration mode.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface port-channel** command to create a new port channel interface at the global configuration level.

```
device(config)# interface port-channel 30
```

NOTE
The port-channel interface ranges from 1 to 1024.

The following example creates a new port channel interface of 30.

```
device# configure terminal
device(config)# interface port-channel 30
```

Deleting a port channel interface

Follow this procedure to delete a port-channel interface and all member interfaces.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. To delete a port-channel interface, enter the **no interface port-channel** command.

```
device(config)# no interface port-channel 30
```

The following example deletes port-channel interface 30.

```
device# configure terminal
device(config)# no interface port-channel 30
```

Adding a member port to a port channel

Follow this procedure to add a port to a specific port channel interface at the interface configuration level. If the port channel is not created, this task creates the port channel and also adds a port to the port channel.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
device(config)#
```

2. Enter the **interface port-channel** command to add a port channel interface at the global configuration level.

```
device(config)# interface port-channel 30
device(conf-Port-channel-30)#
```

3. Configure the **interface ethernet** command to enable the interface.

```
device(conf-Port-channel-30)# interface ethernet 0/5
device(conf-if-eth-0/5)#
```

4. Add a port to the port channel interface as static.

```
device(conf-if-eth-0/5)# channel-group 30 mode on
```

5. Add a port to the port channel interface as a dynamic (using LACP), active or passive mode.

```
device(conf-if-eth-0/5)# channel-group 30 mode active

device(conf-if-eth-0/5)# channel-group 30 mode passive
```

The following example is for a static LAG configuration with the mode ON.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 0/5
device(conf-if-eth-0/5)# channel-group 30 mode on
```

The following example adds a port 0/5 to the existing dynamic port channel interface 30 with the mode active.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 0/5
device(conf-if-eth-0/5)# channel-group 30 mode active
```

The following example adds a port 0/5 to the existing dynamic port channel interface 30 with the mode passive.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 0/5
device(conf-if-eth-0/5)# channel-group 30 mode passive
```

Deleting a member port from a port channel

Follow this procedure to delete a member port from a port channel interface at the interface configuration level.

Delete a port from the port channel interface.

```
device(conf-if-eth-0/5)# no channel-group
```

The following example deletes a port 0/5 from the existing port channel interface 30.

```
device# configure terminal
device(config)# interface ethernet 0/5
device(conf-if-eth-0/5)# no channel-group
```

Configuring the minimum number of LAG member links

Follow this procedure to configure the minimum number of LAG member links that should be functional so that the port-channel interface is operationally up.

This configuration allows a port-channel to operate at a certain minimum bandwidth at all times. If the bandwidth of the port-channel drops below the minimum number, then the port-channel is declared operationally DOWN even though it has operationally UP members.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
device(config)#
```

2. Enter the **interface port-channel** command at the global configuration level.

```
device(config)# interface port-channel 30
device(conf-Port-channel-30)#
```

3. Configure the minimum number of LAG member links at the port-channel interface configuration mode.

```
device(conf-Port-channel-30)# minimum-links 5
```

NOTE
The number of links ranges from 1 to 32. The default minimum links is 1.

The following example sets min-link 5 to the existing port channel interface 30.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# minimum-links 5
```

Configuring the LACP system priority

You configure the LACP system priority on each switch running LACP. LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other switches.

The system priority value must be a number in the range of 1 through 65535. The higher the number, the lower the priority. The default priority is 32768.

To configure the global LACP system priority, perform the following steps:

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Specify the LACP system priority.

```
device(config)# lacp system-priority 25000
```

3. To reset the system priority to the default value.

```
device(config)# no lacp system-priority
```

Configuring the LACP port priority

Follow this procedure to configure the LACP port priority of a member port of a specific port-channel interface.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface port-channel** command to add a port channel interface at the global configuration level.

```
device(config)# interface port-channel 30
device(conf-Port-channel-30)#
```

3. Configure the **interface ethernet** command and add the port to the port-channel interface.

```
device(conf-Port-channel-30)# interface ethernet 0/5
device(conf-if-eth-0/5)# channel-group 30 mode active
```

4. Configure the LACP port priority 12 for the member port.

```
device(conf-if-eth-0/5)# lacp port-priority 12
```

NOTE
The LACP port priority value ranges from 1 to 65535. The default value is 32768.

5. To rest the configured port priority to the default value.

```
device(conf-if-eth-0/5)# no lacp port-priority
```

The example sets the port priority as 12.

```
device# configure terminal
device(config)# interface port-channel 30
device(conf-Port-channel-30)# interface ethernet 0/5
device(conf-if-eth-0/5)# channel-group 30 mode active
device(conf-if-eth-0/5)# lacp port-priority 12
```

Configuring the LACP timeout period

The LACP timeout period indicates how long LACP waits before timing out the neighboring device.

The **short** timeout period is 3 seconds and the **long** timeout period is 90 seconds. The default is **long**. The **short** timeout period specifies that the PDU is sent every second and the port waits three times this long (three seconds) before invalidating the information received earlier on this PDU. The **long** timeout period specifies that the PDU is sent once in 30 seconds and the port waits three times this long (90 seconds) before invalidating the information received earlier on this PDU.

To configure the LACP timeout period on an interface, perform the following steps:

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command, specifying the interface type and the slot/port.

```
device(config)# interface ethernet 0/1
```

3. Enter the **no shutdown** command to enable the interface.
4. Specify the LACP timeout short period for the interface.

```
device(conf-if-eth 0/1)# lacp timeout short
```

5. Specify the LACP timeout long period for the interface.

```
device(conf-if-eth 0/1)# lacp timeout long
```

Configuring LACP default Up

Follow this procedure to activate an LACP link even in the absence of PDUs.

Consider the following when using the **lacp default-up** command:

- The command is available only if the configured interface is a dynamic member of a port-channel interface.
- The command is not supported on static LAGs.
- The command is not supported on port-channel interfaces.

1. Enter the **configure terminal** command to access global configuration mode.

2. Enter the **interface** command, specifying the interface type and the slot/port.

```
device(config)# interface ethernet 0/1
```

3. Specify LACP default-up for the interface.

```
device(conf-if-eth-0/1)# lacp default-up
```

4. Enter the no form of the command to disable the configuration.

```
device(conf-if-eth-0/1)#  no lacp default-up
```

Troubleshooting LACP

To troubleshoot problems with your LACP configuration, use the following troubleshooting tips.

If a standard IEEE 802.1AX-based dynamic trunk is configured on a link and the link is not able to join the LAG, do the following:

- Make sure that both ends of the link are configured as **standard** for the trunk type.
- Make sure that both ends of the link are *not* configured for **passive** mode. They must be configured as **active /active**, **active / passive**, or **passive /active**.
- Make sure that the port-channel interface is in the administrative "up" state by ensuring that the **no shutdown** command was entered on the interface on both ends of the link.
- Make sure that the links that are part of the LAG are connected to the same neighboring switch.
- Make sure that the system ID of the switches connected by the link is unique. You can verify this by entering the **show lacp sys-id** command on both switches.
- Make sure that LACPDUs are being received and transmitted on both ends of the link and that there are no error PDUs. You can verify this by entering the **show lacp counters** *number* command and looking at the receive mode (rx) and transmit mode (tx) statistics. The statistics should be incrementing and should not be at zero or a fixed value. If the PDU rx count is not incrementing, check the interface for possible CRC errors by entering the **show interface** *link-name* command on the neighboring switch. If the PDU tx count is not incrementing, check the operational status of the link by entering the **show interface** *link-name* command and verifying that the interface status is "up."

When a link has problem, the **show port-channel** command displays the following message:

```
Mux machine state: Deskew not OK.
```

If a static trunk is configured on a link and the link is not able to join the LAG, do the following:

- Make sure that both ends of the link are configured as **standard** for trunk type and verify that the mode is "on."
- Make sure that the port-channel interface is in the administrative "up" state by ensuring that the **no shutdown** command was entered on the interface on both ends of the link.

LAG load balancing

This feature allows you to configure the load-balancing feature on a node, to forward traffic. To distribute the traffic among the possible paths, you can configure the load-balancing flavor. Available flavors are listed below.

TABLE 2 Load balancing flavors

Flavor	Definition
dst-mac-vid	Distribution based on Destination MAC address and VLAN ID. (Note that the outer VID is considered).

TABLE 2 Load balancing flavors (continued)

Flavor	Definition
src-mac-vid	Distribution based on source MAC address and VLAN ID.
src-dst-mac-vid	Distribution based on source and destination MAC address and VLAN ID.
src-dst-ip	Distribution based on source and destination IP address.
src-dst-ip-mac-vid	Distribution based on source and destination IP and MAC addresses including the VID.
src-dst-ip-port	Distribution based on source and destination IP addresses and TCP port.
src-dst-ip-mac-vid-port	Distribution based on source and destination IP, MAC address, VLAN, and port.

Load balancing flavor is unique per node and it can be configured at the global configuration mode.

> NOTE
> The default load-balancing flavor is src-dst-ip-mac-vid-port and is applied for all the port channels.

The following example sets the flavor to "destination MAC address and VID-based load balancing."

```
device(config)# load-balance dst-mac-vid
device(config)# exit
device# show running-config load-balance
load-balance dst-mac-vid
device# show port-channel load-balance
 Destination MAC address and VID based load balancing
```

Show and clear LAG commands

This section contains tasks for showing port-channel information and statistics and for clearing the relevant counters.

Displaying port-channel information

Various show commands are used to display information for a port-channel interfaces.

1. Use the **show port-channel summary** command to display brief information of all port-channels.

```
device# show port-channel summary
Flags:  D - Down                P - Up in port-channel (members)
        U - Up (port-channel)   * - Primary link in port-channel
        S - Switched
        M - Not in use. Min-links not met
===== =============== ========== ===============
Group Port-channel    Protocol   Member ports
===== =============== ========== ===============
1     Po 1    (D)     None       Eth 0/25 (D)
                                 Eth 0/26 (D)
2     Po 2    (D)     None       Eth 0/27 (D)
                                 Eth 0/28 (D)
10    Po 10   (U)     LACP       Eth 0/4  (P)
                                 Eth 0/18 (P)
100   Po 100  (U)     None       Eth 0/10 (P)
                                          Eth 0/11 (P)
```

2. Use the **show port-channel detail** command to display detailed information of all the port-channels.

```
device# show port-channel detail
 Static Aggregator: Po 1
 Aggregator type: Standard
 Number of Ports: 2
 Member ports:
   Eth 0/25
   Eth 0/26


 Static Aggregator: Po 2
 Aggregator type: Standard
 Number of Ports: 2
 Member ports:
   Eth 0/27
   Eth 0/28


 Static Aggregator: Po 100
 Aggregator type: Standard
 Number of Ports: 2
 Member ports:
   Eth 0/10
   Eth 0/11


 LACP Aggregator: Po 10
Aggregator type: Standard
  Actor System ID - 0x8000,76-8e-f8-0a-98-00
  Admin Key: 0010 - Oper Key 0010
  Receive link count: 2 - Transmit link count: 2
  Individual: 0 - Ready: 1
  Partner System ID - 0x8000,76-8e-f8-0a-68-00
  Partner Oper Key 0010
 Number of Ports: 2
 Member ports:
   Link: Eth 0/4 (0x18820016) sync: 1
   Link: Eth 0/18 (0x18890084) sync: 1
```

3. Use the **show port-channel** *number* command to display detailed information of a specific port-channel interface

```
device# show port-channel 10
LACP Aggregator: Po 10
 Aggregator type: Standard
  Admin Key: 0010 - Oper Key 0010
  Partner System ID - 0x8000,76-8e-f8-0a-68-00
  Partner Oper Key 0010
 Number of Ports: 2
 Member ports:
   Link: Eth 0/4 (0x18820016) sync: 1
   Link: Eth 0/18 (0x18890084) sync: 1
```

Displaying LAG hashing

Use the **show port-channel load-balance** command to display the configured parameters for LAG hashing.

```
device# show port-channel load-balance
Header parameters
        Ethernet Mask: sa-mac da-mac etype vlan
        ip: src-ip dst-ip protocol src-l4-port dst-l4-port
        ipv6: ipv6-src-ip ipv6-dst-ip ipv6-next-hdripv6-src-l4-port ipv6-dst-l4-port

Hash Settings
        hdr-start:FWD, hdr-count:1, bos-start:0, bos-skip:0, skip-cw:0
        normalize:0, rotate:3, include_src_port:0, Disable: L2 0, ipv4 0, ipv6 0
```

```
load-balance-type hash-based
```

Displaying LACP system-id information

Follow this procedure to display LACP system ID and priority information.

Enter the **show lacp sys-id** command to display LACP information for the system ID and priority.

```
device# show lacp sys-id
System ID: 0x8000,76-8e-f8-0a-98-00
```

Displaying LACP statistics

Follow this procedure to display LACP statistics for a port-channel interface or for all port-channel interfaces.

Enter the **show lacp counters** command to display LACP statistics for a port-channel.

```
device# show lacp counter
Traffic statistics
Port     LACPDUs   Marker   Pckt err Sent  Recv  Sent  Recv  Sent  Recv Aggregator
Eth 0/6   110                        0        0     0        0     0         Po 3
```

Clearing LACP counter statistics on a LAG

This topic describes how to clear LACP counter statistics on a single LAG.

Enter the **clear lacp** *LAG_group_number* **counters** command to clear the LACP counter statistics for the specified LAG group number.

```
device# clear lacp 42 counters
```

Clearing LACP counter statistics on all LAG groups

This topic describes how to clear the LACP counter statistics for all LAG groups.

Enter the **clear lacp counter** command to clear the LACP counter statistics for all LAG groups.

```
device# clear lacp counter
```

VLANs

- 802.1Q VLAN overview 25
- Configuring VLANs 25
- VLAN statistics 30
- Configuring the MAC address table and conversational MAC learning 32
- Configuring virtual routing interfaces 34
- DHCP Layer 2 relay agent Option 82 for VLANs 34
- Endpoint tracking 39

802.1Q VLAN overview

IEEE 802.1Q VLANs provide the capability to overlay the physical network with multiple virtual networks. VLANs allow you to isolate network traffic between virtual networks and reduce the size of administrative and broadcast domains.

A VLAN contains end stations that have a common set of requirements that are independent of physical location. You can group end stations in a VLAN even if they are not physically located in the same LAN segment. VLANs are typically associated with IP subnetworks and all the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. VLAN membership is configurable on a per-interface basis.

Configuring VLANs

The following sections discuss working with VLANs on Extreme devices.

Configuring a VLAN

Follow this procedure to configure a VLAN in the Extreme device at the global configuration level.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **vlan** command to create a topology group at the global configuration level.

```
device(config)# vlan 5
device(config-vlan-5)#
```

NOTE
The **no vlan** command removes the existing VLAN instance from the device.

Configuring a switchport interface

Follow this procedure to configure a switchport interface in the device to send and receive data packets.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to configure the interface mode.

```
device(config)# interface ethernet 0/1
```

3. Enter the **switchport** command to configure a switchport interface.

```
device(conf-if-eth-0/1)# switchport
```

Configuring the switchport interface mode

Do the following to set the switchport interface as access or trunk. This configuration works only when the interface is set as switchport.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to configure the interface mode.

```
device(config)# interface ethernet 0/1
```

3. Enter the **switchport** command to set the interface as switchport.

```
device(conf-if-eth-0/1)# switchport
```

4. Enter the **switchport mode** command to configure the switchport interface in trunk mode.

```
device(conf-if-eth-0/1)# switchport mode trunk
```

NOTE
The default mode is access. Enter the **switchport mode access** command to set the mode as *access*.

NOTE
Before you change the switch port mode from **switchport mode access** with an explicit **switchport access vlan** to **switchport mode trunk-no-default-native**, you must enter the **no switchport** command on the interface level, and then enter the **switchport** command to set the interface as a switchport. Now you can configure the **switchport mode trunk-no-default-native** command.

Configuring the switchport access VLAN type

Do the following to change the switchport access VLAN type. This configuration works only when the interface is set as switchport.

Ensure that reserved VLANs are not used. Use the **no switchport access vlan** command to set the default VLAN as the access VLAN.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to specify an Ethernet interface.

```
device(config)# interface ethernet 0/1
```

3. Enter the **switchport** command to set the interface as switchport.

```
device(conf-if-eth-0/1)# switchport
```

4. Enter the **switchport access vlan** command to set the mode of the interface to *access* and specify a VLAN.

```
device(conf-if-eth-0/1)# switchport access vlan 10
```

This example sets the mode of a specific port-channel interface to *trunk*.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-port-channel-35)# switchport mode trunk
```

Configuring a VLAN in trunk mode

Do the following to add or remove VLANs on a Layer 2 interface in trunk mode. The configuration is also used to configure the VLANs to send and receive data packets.

Ensure that reserved VLANs are not used.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to specify an Ethernet interface.

```
device(config)# interface ethernet 0/1
```

3. Enter the **switchport** command to set the interface as switchport.

```
device(conf-if-eth-0/1)# switchport
```

4. Enter the **switchport trunk allowed vlan** command to set the mode of the interface to *trunk* and add a VLAN.

```
device(conf-if-eth-0/1)# switchport trunk allowed vlan add 5
```

The example sets the mode of the Ethernet interface to *trunk*.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# switchport mode trunk
```

The example sets the mode of a port-channel interface to *trunk* and allows all VLANs.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-Port-channel-35)# switchport trunk allowed vlan all
```

Configuring a native VLAN on a trunk port

Do the following to set native VLAN characteristics on a trunk port for classifying the untagged traffic data packets.

Ensure that reserved VLANs are not used.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
device(config)#
```

2. Enter the **interface ethernet** command to configure the interface mode.

```
device(config)# interface ethernet 0/1
```

3. Enter the **switchport** command to set the interface as switchport.

```
device(conf-if-eth-0/1)# switchport
```

4. Enter the **switchport trunk native-vlan** command to set native VLAN characteristics to *access* and specify a VLAN.

```
device(conf-if-eth-0/1)# switchport trunk native-vlan 300
```

This example removes the configured native VLAN on the Ethernet interface.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# no switchport trunk native-vlan 300
```

Enabling VLAN tagging for native traffic

Do the following to enable tagging for native traffic on a specific interface.

Ensure that reserved VLANs are not used.

The following table describes the acceptable frame types, as well as system behavior, for tagged native VLAN, untagged native VLAN, and no native VLAN.

TABLE 3 Acceptable frame types and system behavior for native VLANs

	Tagged native VLAN	Untagged native VLAN	No native VLAN
Configuration	switchport trunk tag native-vlan (Default) **and** Globally: **vlan dot1q tag native**	no switchport trunk tag native-vlan **or** Global config: **no vlan dot1q tag native**	switchport mode trunk-no-default-native
Acceptable frame type	VLAN-tagged only	All (tagged and untagged)	VLAN-tagged only
Receive untagged	Drop	Forward/flood in native VLAN	Drop
Receive tagged on native VLAN	Forward/flood in native VLAN	Forward/flood in native VLAN	Drop
Transmit on native VLAN	Tagged with native VLAN	Untagged packet	Will not forward on native VLAN

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to configure the interface mode.

```
device(config)# interface ethernet 0/1
```

3. Enter the **switchport** command to set the interface as switchport.

```
device(conf-if-eth-0/1)# switchport
```

4. Enter the **switchport trunk tag native-vlan** command to enable tagging for native traffic data VLAN characteristics on a specific interface.

```
device(conf-if-eth-0/1)# switchport trunk tag native-vlan
```

This example enables tagging for native traffic data on a specific Ethernet interface.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# switchport trunk tag native-vlan
```

This example disables the native VLAN tagging on a port-channel.

```
device# configure terminal
device(config)# interface port-channel 35
device(config-Port-channel-35)# no switchport trunk tag native
```

Displaying the status of a switchport interface

Do the following to display detailed Layer 2 information for all switchport interfaces.

Enter the **show interface switchport** to display the detailed Layer 2 information for all interfaces.

```
device# show interface switchport
 Interface name          : Eth 0/1
 Switchport mode         : access
 Ingress filter          : enable
 Acceptable frame types  : all
 Default Vlan            : 1
 Active Vlans            : 1
 Inactive Vlans          : -
 Interface name          : Port-channel 5
 Switchport mode         : access
 Ingress filter          : enable
 Acceptable frame types  : all
 Default Vlan            : 1
 Active Vlans            : 1
```

Displaying the switchport interface type

Do the following to display detailed Layer 2 information for a specific interface.

Enter the **show interface switchport** to display the detailed Layer 2 information for a specific interface.

```
device# show interface ethernet 0/1 switchport
 Interface name           : ethernet 0/1
 Switchport mode          : trunk
 Fcoeport enabled         : no
 Ingress filter           : enable
 Acceptable frame types   : vlan-tagged only
 Native Vlan              : 1
 Active Vlans             : 1,5-10
 Inactive Vlans           : -
```

The example displays the detailed Layer 2 information for a port-channel interface.

```
device# show interface port-channel 5 switchport
 Interface name           : Port-channel 5
 Switchport mode          : access
 Fcoeport enabled         : no
 Ingress filter           : enable
 Acceptable frame types   : vlan-untagged only
 Default Vlan             : 1
 Active Vlans             : 1
 Inactive Vlans           : -
```

Verifying a switchport interface running configuration

Do the following to display the running configuration information for the Layer 2 properties for a specific interface.

Enter the **show running-config interface** to display the running configuration information for a specific interface.

```
device# show running-config interface ethernet 0/1 switchport
interface interface Eth 0/1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 5-10
 switchport trunk tag native-vlan
```

This example displays the running configuration information for a port-channel interface.

```
device# show running-config interface port-channel 5 switchport
interface Port-channel 5
 switchport
 switchport mode access
 switchport access vlan 1
```

Displaying VLAN information

Do the following to display information about a specific VLAN.

Enter the **show vlan** to display information about VLAN 1.

```
device# show vlan 1
VLAN Name State Ports
(u)-Untagged, (t)-Tagged
(c)-Converged
==========================================================
1 default ACTIVE Eth 0/1(t) Eth 0/4(t) Eth 0/5(t) Eth 0/8(t)
```

VLAN statistics

Devices gather statistics for all ports and port channels on configured VLANs.

Use the **statistics** command in the VLAN configuration mode to enable statistics on a VLAN.

NOTE
Statistics has to be manually enabled for a specific VLAN, since it is not enabled by default for VLANs.

Please note that:

- The statistics reported are not real-time statistics since they depend upon the load on the system.
- Statistics has to be manually enabled for a specific VLAN. This ensures better utilization of the statistics resources in the hardware.
- Statistics for VLANs with VE interfaces consider only the switched frames. Packets which are routed into or out of the VE interface are not counted.
- Enabling statistics on a VLAN has a heavy impact on the data traffic.

Enabling statistics on a VLAN

Follow this procedure to enable statistics on a VLAN.

1. Enter the global configuration mode.

```
device# configure terminal
```

2. Enter the **vlan** command to specify a VLAN for statistics collection.

```
device(config)# vlan 5
device(config-vlan-5)#
```

3. Enter the **statistics** command to enable statistics for all ports and port channels on configured VLANs.

```
device(config-vlan-5)# statistics
```

NOTE

Use the **no statistics** command to disable statistics on VLANs.

```
device(config-vlan-5)# no statistics
```

Displaying statistics for VLANs

Do the following to display statistics information for VLANs.

Enter the **show statistics vlan** command to view the statistics for all ports and port channels on all configured VLANs.

```
device# show statistics vlan

Vlan 10 Statistics
Interface     RxPkts          RxBytes          TxPkts          TxBytes
eth 0/1       821729          821729           95940360        95940360
eth 0/2       884484          885855           95969584        95484555
po  1        8884            8855              9684            9955

Vlan 20 Statistics
Interface     RxPkts          RxBytes          TxPkts          TxBytes
eth 0/6       821729          821729           95940360        95940360
eth 0/21     8884            8855              9684            9955
po  2      884484          885855           95969584        95484555
```

TABLE 4 Output descriptions of the show statistics vlan command

Field	Description
Interface	The interface whose counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
RxBytes	The number of bytes received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying VLAN statistics for a specific VLAN

Enter the **show statistics vlan** *vlan ID* command to view the statistics for a specific VLAN. Here *vlan ID* is the specific VLAN ID.

```
device# show statistics vlan 10

Vlan 10 Statistics
Interface     RxPkts          RxBytes          TxPkts          TxBytes
```

```
eth 0/1       821729           821729           95940360         95940360
eth 0/2       884484           885855           95969584         95484555
po  1      8884             8855             9684             9955
```

Clearing statistics on VLANs

Follow the procedure to clear statistics' information for VLANs.

Enter the **clear statistics vlan** command to clear the statistics for all ports and port channels on all configured VLANs.

```
device# clear statistics vlan
```

Clearing statistics for a specific VLAN

Enter the **clear statistics vlan** *vlan ID* command to clear the statistics for a specific VLAN. Here *vlan ID* is the specific VLAN ID.

```
device# clear statistics vlan 10
```

Configuring the MAC address table and conversational MAC learning

Each DCB port has a MAC address table that stores the source MAC address of all frames. In addition, there is a configurable aging timer. If a source MAC address remains inactive for a specified number of seconds, it is removed from the address table.

Conversational MAC learning

Layer 2 switches use forwarding tables to direct traffic to specific ports, based on the VLAN number and destination MAC address of the frame.

When there is no entry corresponding to the destination MAC address in the incoming VLAN, the frame is sent to all forwarding ports within the respective VLAN, which causes flooding. MAC address learning is an essential Layer 2 feature whereby the source MAC addresses of each received packet is stored so that future packets destined for that address can be forwarded only to the bridge interface on which the address is located.

Using the global **mac-address-table** command with the **conversational** keyword enables conversational MAC (address) learning, or CML, globally on a switch.

Specifying or disabling the aging time for MAC addresses

You can set the length of time that a dynamic entry remains in the MAC address table after the entry is used or updated. Static address entries are never aged or removed from the table. You can also disable the aging time. The default is 1800 seconds.

> NOTE
> To disable the aging time for MAC addresses, enter an aging time value of 0.

Do the following to specify an aging time or disable the aging time for MAC addresses.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. The following example specifies an aging-time of 600 seconds.

```
device(config)# mac-address-table aging-time 600
```

3. Enter the **no mac-address-table aging-time** command to restore the default aging time.

```
device(config)# no mac-address-table aging-time
```

The maximum value supported is 100000 seconds.

NOTE: There may be a short delay in the configured MAC aging time. The hardware scans the MAC entries every 1/7th of the aging time. For example, if the configured aging time is 300 seconds, there can be a delay of 43 seconds. In a scaled environment, the hardware sends only 16 K aging events for every aging cycle.

Adding static addresses to the MAC address table

Do the following to add a static address to the MAC address table.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **mac-address-table** command with **static** keyword to configure the static address 0011.2222.3333 to the MAC address table, for a packet received on VLAN 100 on a Ethernet interface, as in the following example.

```
device(config)# mac-address-table static 0011.2222.3333 forward ethernet 0/1 vlan 100
```

Enabling conversational MAC learning (CML)

You can enable conversation-based MAC address learning by means of the **mac-address-table learning-mode conversational** command.

ATTENTION
The ability to disable source MAC address learning on a per-port, per-VLAN basis constrains traffic flooding to only the ports that are part of a VLAN. Disabling traditional dynamic MAC learning prevents the MAC address table from being saturated. For example, when a device is being attacked by many packets with different source MAC address, the updating of the MAC address table is significantly impaired.

NOTE
For the CML scale supported, refer to the Release Notes.

Do the following in global configuration mode to enable CML globally.

```
device(config)# mac-address-table learning-mode conversational
```

Do the following to revert to legacy dynamic MAC learning mode.

```
device(config)# no mac-address-table learning-mode conversational
```

Do the following to configure the destination MAC address aging interval to 60 seconds.

```
device(config)# mac-address-table aging-time conversational 60
```

Do the following to revert to the default conversational aging interval of 300 seconds.

```
device(config)# no mac-address-table aging-time conversational
```

Configuring virtual routing interfaces

The Extreme device sends Layer 3 traffic at Layer 2 within a protocol-based VLAN. However, Layer 3 traffic from one protocol-based VLAN to another must be routed. If you want the device to be able to send Layer 3 traffic from one protocol-based VLAN to another on the same device, you must configure a virtual routing interface on each protocol-based VLAN, then configure routing parameters on the virtual routing interfaces.

A *virtual routing interface* is a logical routing interface that the Extreme device uses to route Layer 3 protocol traffic between protocol-based VLANs. It is a logical port on which you can configure Layer 3 routing parameters.

For example, to enable a Extreme device to route IP traffic from one IP protocol VLAN to another, you must configure a virtual routing interface on each IP protocol VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

To attach a router interface to a VLAN, using the **router-interface** command:

```
device# configure terminal
device(config)# vlan 2
device(config-vlan-2)# router-interface ve 2
```

NOTE

Only one router VE interface can be mapped to a VLAN. The VLAN ID and the VE ID need not be the same.

Use the **no router interface ve** command to remove the router VE interface.

DHCP Layer 2 relay agent Option 82 for VLANs

This feature supports DHCP Relay Agent Information Option 82 for Layer 2 on VLANs.

NOTE

For a discussion of this feature for Layer 3, refer to the "DHCPv4" chapter in the *Extreme SLX-OS Layer 3 Configuration Guide*.

Overview

It is advantageous to have DHCP relay agent support when DHCP clients and servers are not on the same subnet. DHCP broadcast requests are relayed by the DHCP relay agent to the DHCP server. The DHCP replies are unicasted to the DHCP relay agent, which in turn relays them back to the DHCP client.

For Layer 3, relay agents populate the GIADDR (global IP address) field and also append the "Relay Agent Information" option. DHCP servers use this option for assigning the IP address and other parameters.

In some network configurations that have Layer 2 devices between the DHCP client and the DHCP relay agent, it is better to have Layer 2 relay agents running on Layer 2 devices. This places the agents in a better position to append the Relay Agent Information option because they are closer to the DHCP clients. These agents also broadcast the DHCP messages. (A Layer 3 relay agent, on the other hand, relays the information to the DHCP server.)

Option 82

When this Layer 2 feature is enabled on a VLAN, the Option 82 information is inserted by the Layer 2 relay agent before the DHCP messages are broadcasted further. This information allows the DHCP server to select an IP address or other parameter. The DHCP server echos Option 82 in the reply packets. The DHCP Layer 2 relay agent validates and removes Option 82 and sends the response to the DHCP client.

The format of Option 82 is shown in the following table.

TABLE 5 Option 82 format

Code	Len	Agent Information Field					
82	N	i1	i2	i3	i4	...	iN

NOTE

The length *N* represents the total number of octets in the Agent Information Field. The Agent Information field consists of a sequence of SubOpt/Length/Value tuples for each suboption.

The following table lists the circuit ID suboptions that are added by the relay agent.

TABLE 6 Circuit ID suboptions

Suboption type (1 byte)	Length (1 byte)	VLAN ID <string> (4 bytes)	IF-description string (4 bytes)
2	68		

NOTE

The circuit ID is a combination of the VLAN ID and the interface description string. If the interface description is not configured, the default string "Extreme" is used in the circuit ID.

The following table lists the remote ID suboptions that are added by the relay agent.

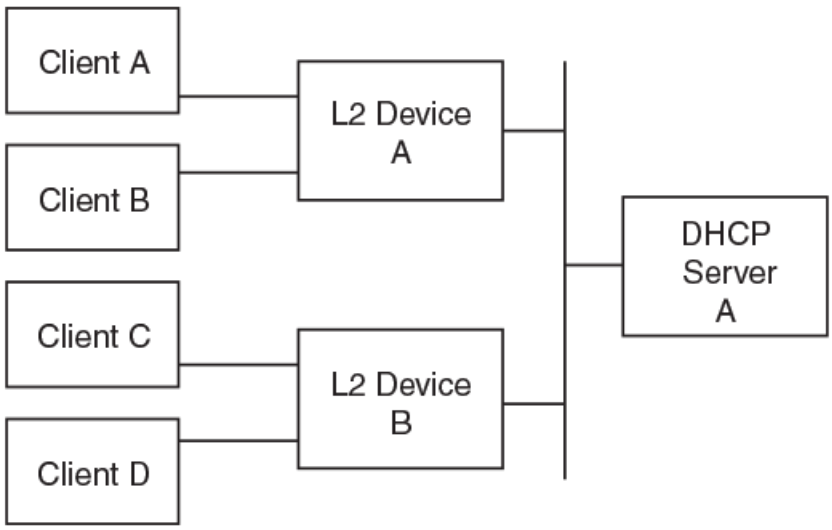
TABLE 7 Remote ID suboptions

Suboption type (1 byte)	Length (1 byte)	VLAN ID (2 bytes)	MAC address (6 bytes)
2	8		

Topologies and operation

The following figure illustrates a single DHCP server and clients on the same subnet.

FIGURE 1 Single DHCP server and clients on same subnet

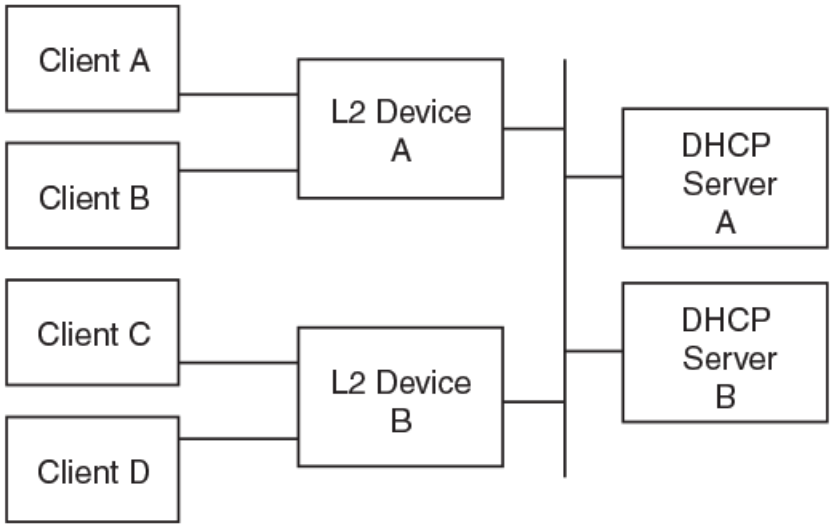


Note the following considerations for the above scenario:

- The Layer 2 relay agent (on the Layer 2 device) appends Option 82 information to a DHCP message received from the client and broadcasts the message to all other ports. The Option 82 information contains the remote ID suboption and the circuit ID suboption.
- The Layer 2 relay agent does not set the GIADDR field.
- The DHCP server echoes the Relay Agent Information option in the response message.
- The Layer 2 relay agent uses the Relay Agent Information option to find out whether it had appended Option 82 to the request message.
- The Layer 2 relay agent removes Option 82 from the packet received from the DHCP server after this option is validated, and then it forwards the message to the interface identified by the Relay Agent Information option.

The following figure illustrates multiple DHCP servers and clients on the same subnet.

FIGURE 2 Multiple DHCP servers and clients on the same subnet

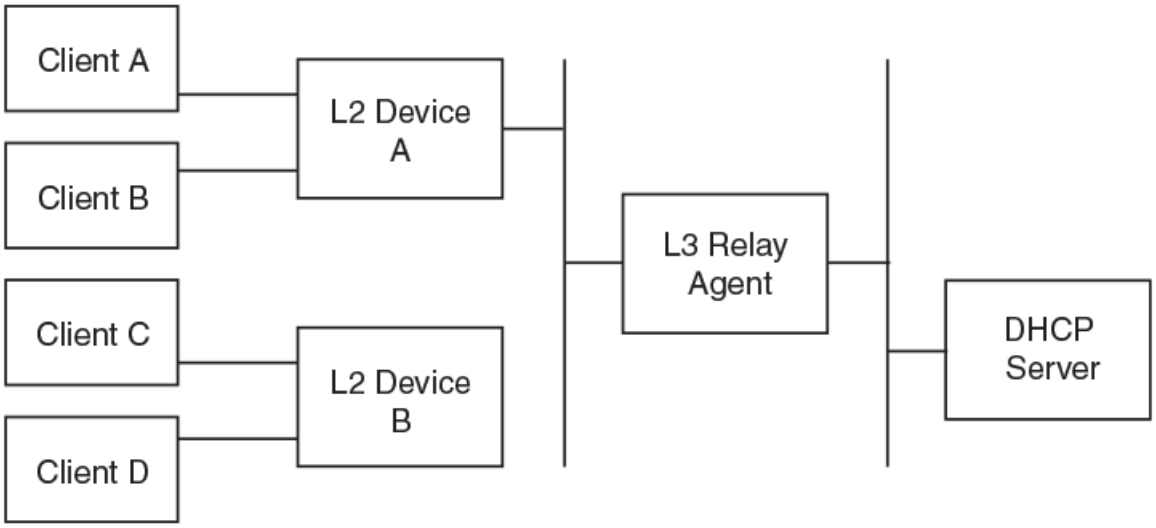


Note the following considerations for the above scenario:

- The Layer 2 relay agent receives multiple responses (from all the servers).
- The processing of DHCP messages in the Layer 2 relay agent remains the same as in the scenario above.

The following figure illustrates a DHCP server on another subnet with clients and one Layer 3 relay agent.

FIGURE 3 DHCP server on another subnet with clients and one Layer 3 relay agent



Note the following considerations for the above scenario:

- The Layer 3 relay agent receives the message relayed by the Layer 2 relay agent. It finds that the message already contains a Relay Agent Information option. The Layer 3 relay agent populates the GIADDR field as appropriate and relays the message to the DHCP server.

- The DHCP server processes the message and unicasts the response to the Layer 3 relay agent on the address specified in the GIADDR field.
- The L3 relay agent processes the response from the server and identifies the outgoing interface. It resets the GIADDR field and broadcasts the message on the identified outgoing interface.

Considerations for Layer 2 relay agent

Note the following:

- All client interfaces are treated as untrusted interfaces.
- There are no separate configurations for the suboptions. Enabling Option 82 for this feature enables the automatic insertion of both the circuit ID and remote ID suboptions.

Configuring DHCP Layer 2 relay agent Option 82

This task enables, verifies, and disables DHCP Relay Agent Information Option 82 for Layer 2 on VLANs.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Specify a VLAN and enter VLAN configuration mode.

```
device(config)# vlan 10
```

3. Enter the **t2 relay agent information option** command.

```
device(config-vlan-10)# l2 relay agent information option
DHCP L2 Relay Agent Information Option is enabled.
```

4. To verify the running configuration on the VLAN, enter the **show running-config** command and specify the VLAN.

```
device# show running-config vlan 10
    vlan 10
    ip dhcp relay information option
```

5. To verify the configuration on the switch, enter the **show ip dhcp relay option** command.

```
device# show ip dhcp relay option
Interface      Circuit-ID               Remote-ID
---------      ----------               ---------
eth0/1         0201630080Extreme        000a0027f8c744e4
eth0/2         0201630081EdgeDevices    00140027f8c744e5
po1000         0201630082Clusters       004b0027f8c744e9
ve10           0201630097Extreme        00640027f8c744f1
```

   You can use the **interface** keyword in the above command to specify an interface.

6. To disable this feature, enter the **no t2 relay agent information option** command.

```
device(config-vlan-10)# no l2 relay agent information option
DHCP L2 Relay Agent Information Option is disabled.
```

Endpoint tracking

The endpoint tracking feature minimizes the configuration and management of VLANs on switches in the data center, by means of MAC authentication using RADIUS protocol.

Overview

NOTE
This feature is supported only on the SLX 9140.

Statically provisioning VLANs in a data center network has the following drawbacks:

- Managing VLANs on top-of-rack (TOR) switches is tedious for the administrator.
- Having VLANs provisioned ahead of time increases the size of the active topology for control protocols such as STP and RSTP, increasing convergence times.
- Flood (unknown unicast, broadcast) traffic can unnecessarily eat up bandwidth on the TOR-to-EOR (end of row) links.
- For the case when the virtual machine (VM) sends or receives tagged traffic, flood traffic can consume CPU cycles on every server that is connected to the network.
- The dynamic VLAN feature allows SLX-OS to create, prune, and open VLANs on the switch dynamically as they are needed by the VMs. This enables the VLAN to follow the VM as it migrates between servers in the data center.

This feature provides the following to remedy the above drawbacks:

- Association of MAC addresses to specific VLANs.
- Once a MAC address is authorized and the VLAN is not already provisioned on the switch, the dynamic (1) creation of the VLAN to which this MAC is associated, (2) configuration of the tag, and (3) addition of the feature-enabled port on which the MAC was detected.
- Once the last MAC address using the VLAN is deleted or aged out, deletion of the VLAN and associated resources on the switch.

The endpoint tracking feature also authorizes the VM. When a VM (and MAC address) is authorized, SLX-OS dynamically creates the VLAN that is required for the VM to send traffic. If a VM shuts down or is moved, its VLAN is pruned to preserve bandwidth. In this way the network responds to changes in the VM network.

Additional details and considerations

Note the following:

- The user enables or disables this feature on a Layer 2 port (switchport), by means of the **endpoint-tracking enable** command. This feature is also supported on LAG and MCT ports.
- SLX-OS communicates with a RADIUS server or XMC-NAC (Extreme Management Center Network Access Control) for MAC authentication information, which can map a MAC address to a VLAN. RADIUS VLAN assignment triggers the creation of the VLAN and the port VLAN membership. SLX-OS expects MAC-to-VLAN binding information from RADIUS. VNI (VXLAN Network Identifier) assignment is not supported.
- The maximum number of authenticated or nonauthenticated MAC addresses supported for each endpoint-tracking-enabled port is 8000 per system.
- Port VLAN membership is local to SLX-OS. The removal of port VLAN membership or the deletion of the VLAN is not propagated to the RADIUS server.

- On an access port, if RADIUS does not provide MAC-to-VLAN mapping, SLX-OS assigns a default VLAN. Only the first RADIUS VLAN assignment is honored, and subsequent MAC addresses cannot override the existing assignment.
- For VMs sending tagged traffic, if the RADIUS authentication response does not have a VLAN assignment for the MAC, SLX-OS creates the VLAN (assuming the VLAN does not already exist) with the received packet's tag and adds the port to the VLAN as tagged. Dynamically created VLANs are not saved across reboots, and therefore are not part of the running configuration.
- When a port becomes part of a VLAN dynamically, this is not reflected in the running configuration of the interface.
- In reauthentication, RADIUS sends two attributes as part of a Change of Authorization (CoA) request (a CoA is an unsolicited message from RADIUS to the switch to trigger an authentication-related action): (1) Calling-station-id, a value field that holds the MAC being authenticated, and VSA, a vendor-specific attribute value field that holds commands and parameters for commands. RADIUS matches the outstanding requests response by means of a CoA identifier.

RADIUS VSA support

This feature supports the following VSA format.

TABLE 8 VSA format

Type (26)	Length	Vendor ID (Extreme Vendor ID)	
Vendor ID (cont'd)		Sub-attr (Sub-type)	Sub-length
Value			

The VSA fields are as follows.

TABLE 9 VSA fields

Field	Description
Type	8-bit field. Always 26 as it represents a VSA attribute.
Length	8-bit field. Length of entire attribute, including type and length fields.
Vendor ID	4 octets, encoding the Extreme Vendor ID.
Sub-attr (Sub-type)	8-bit field, indicating class of command.
Sub-length	8-bit field. Length of the Sub-attr (attribute), including sub-type and sub-length fields.
Value	A string, integer, or IP address based on Sub-type

As a RADIUS Access response, only one VSA attribute is provided, for Egress-vlan. The Egress-vlan VSA format has sub-type 216 and a value field of type integer. The format is shown below.

TABLE 10 Egress-vlan VSA format

Type (26)	Length	Vendor ID (Extreme Vendor ID)	
Vendor ID (cont'd)		Sub-type (216)	Sub-length
egress VLAN_ID			

NOTE
Both Cisco and Extreme Networks Vendor IDs are supported for the above.

For MAC reauthentication, RADIUS sends a Change of Authorization (CoA) of Code 43 with the VSA. This has a sub-type of 1 and the Value field is the string "subscriber:command=reauthenticate". The format is shown below.

TABLE 11 MAC reauthentication format

Type (26)	Length	Vendor ID (Extreme Vendor ID)	
Vendor ID (cont'd)		Sub-type (1)	Sub-length
subscriber:command=reauthenticate			

NOTE

Both Cisco and Extreme Networks Vendor IDs are supported for the above.

VXLAN Extension Tunnels

- VXLAN extension tunnels overview....43
- Configuring VXLAN extension tunnels....46
- Configuring logical VTEPs....47

VXLAN extension tunnels overview

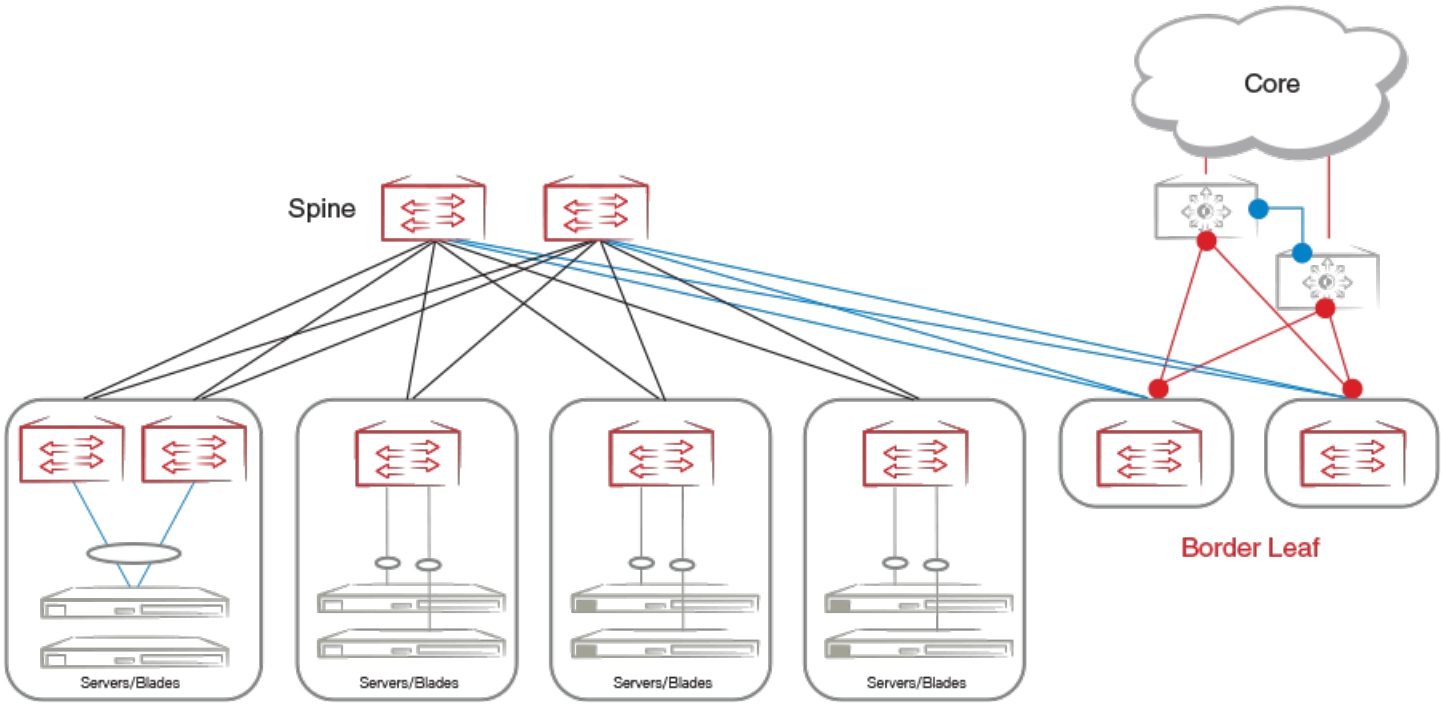
Virtual eXTensible LAN (VXLAN) is an overlay technology that is used to extend Layer 2 domains over Layer 3 networks.

This technology uses existing data center equipment to achieve the extension, providing the benefits of Layer 3, for example, multipathing and distance dependence, while achieving Layer 2 multitenancy. Extension is used to provide Layer 2 overlay services within an IP Fabric, with or without the use of BGP EVPN.

VXLAN tunnel endpoints (VTEPs) support frame encapsulation, where the VTEP has uplink and downlink logical interfaces. The uplink interface receives VXLAN frames and acts as a tunnel endpoint with an IP address to route the VXLAN-encapsulated frames. VXLAN tunnels can be either configured manually, or autodiscovered by means of BGP. An overlay gateway defines a VTEP. Only one overlay gateway is supported per system.

The following figure depicts a sample topology.

FIGURE 4 Sample topology



The spine nodes are SLX 9240 platforms, and the leaf nodes are SLX 9140 platforms. Connection from border leaf nodes to the core is through SLX 9850 series platforms. Layer 2 traffic is within the data center, with Layer 2 extension beyond the data center to the core. The SLX 9140 provides VXLAN-based Network Virtualization Edge (NVE) extension, with NVE redundancy achieved through logical

VXLAN tunnel endpoints (VTEPs). Logical VTEP (LVTEP) spanning supported for up to two nodes to achieve this redundancy, with traffic forwarded an active-active manner.

Layer 2 multitenancy is achieved by means of VLANs and bridge domains (BDs) over the extension tunnels. Both user-configured VLANs and BDs (VLANs/BDs) map to an internal VLAN ID (IVID).

Both loopback and VRRP interfaces are sources for VTEP. The interfaces connecting to the spine are configured as unnumbered interfaces or router ports. The VXLAN tunnels are not terminated at a border leaf; instead, the border leaf nodes act as underlay IP routers. Tunnels can be created manually, through the CLI, or automatically, by means of BGP.

Support is provided for data-plane– and BGP–based MAC address learning and aging on tunnel interfaces. Broadcast, unknown multicast, and multicast (BUM) traffic is flooded on all tunnels, and split horizon is used to prevent loops.

The following sections provide additional details.

Layer 2 data path

Packets received on local or edge ports are processed as regular Layer 2 unicast/BUM packets. Packets that are forwarded over the tunnel, either as a result of a lookup success in the forwarding database (FDB) or as a result of BUM forwarding, have their VXLAN headers encapsulated.

In the case of LVTEP, known unicast packets are forwarded over the local tunnel if that tunnel is operationally up, or over an Inter-Chassis Link (ICL) if the remote tunnel is operationally up. BUM traffic is flooded to the flood list for the IVID associated to the packet. The flood list contains the ICL, the local edge ports, and the local tunnel (if that tunnel has been elected as the designated forwarder by the control plane). Known multicast packets are flooded to multicast group members associated with the packet. The multicast group contains the ICL and other members derived by the control plane.

Packets received over the tunnels are Layer 2 terminated, and the overlay is processed when the outer MAC address matches the MAC of the system and the outer destination IP (DIP) address matches the VTEP IP address. (Tunnel termination is not VRF aware.) If a packet is tunnel terminated and there is no corresponding IVID configured (by the **map** command) for the VXLAN Network Identifier (VNI), the packet is dropped.

Dataplane MAC learning is supported only for CLI-configured extension tunnels. It is not supported for BGP-created tunnels. The MAC entries are aged as they are for physical ports and port channels. Extension tunnels that are BGP-created, or CLI-created with the MAC learning mode as BGP, have their MAC addresses populated by means of BGP EVPN.

Known unicast frames are forwarded according the result of destination lookup. For LVTEP, if the lookup results in a local port, the packet is forwarded to the local port. If the lookup results in a single-homed port that is attached to the peer, then the packet is forwarded over the ICL. If the lookup results in a Multi-Chassis Trunk (MCT), then the packet is forwarded to the local members of the MCT if the local member is operationally up; otherwise, the packet is forwarded to the peer over the ICL.

BUM traffic is flooded on all members of the IVID except the source. For LVTEP, packets are flooded to the flood list for the IVID associated with the packet. The list contains the ICL and the local single-homed edge ports, and local MCT member ports if they are elected as designated forwarders.

For known multicast packets the behavior is similar to that for BUM, except that the flood list contains the members of the mulltiicast group instead of all members of the VLAN.

Split horizon ensures that packets received from a tunnel in one VTEP are not forwarded to another other tunnel in the local or remote VTEP peer if both tunnels belong to the same VXLAN domain.

Layer 3 data path

In Layer 2 extension mode, routing both into and out of tunnels is supported. The Layer 3 interface is on a virtual Ethernet (VE) on an extended VLAN. Both IPv4 and IPv6 underlay routing is supported. Connected subnet and static-route forwarding is supported over tunnels. (The is no user control through CLI over the tunnel maximum transmission unit (MTU). The user must accommodate the increase in packet size that results from encapsulation. Packets that violate the outgoing interface MTU after encapsulation are dropped.

Logical VTEP control plane

BGP EVPN is used as the control plane to achieve LVTEP functionality. MAC syncing, designated forwarder election, and failovers are all achieved through BGP EVPN.

VXLAN resiliency

Tunnel path resiliency is achieved through Layer 3 underlay multipathing. For tunnel origination, the outgoing Layer 3 interface and the next-hop router are determined by the routing information base (RIB). The outgoing Layer 2 interface is determined by ARP and MAC procedures on the next-hop router. ECMP is used to load-balance traffic across available equal-cost paths. The failure of one path causes traffic to be redistributed on the remaining paths, resulting in traffic loss. The availability of a new path results in traffic being redistributed to that path.

Tunnel underlay resiliency

Tunnel path resiliency is achieved through Layer 3 underlay multipathing. For tunnel origination, the outgoing Layer 3 interface and the next-hop router are determined by the routing information base (RIB). The outgoing Layer 2 interface is determined by the ARP and MAC procedures on the next-hop router.

Dynamic changes to the outgoing Layer 3 interface, next-hop router, and the outgoing Layer 2 interface can result from Layer 3 routing path changes or underlying Layer 2 network changes.

In the case of ECMP, traffic is load balanced across the available equal-cost paths. The load balancing algorithm is the same as that used for Layer 3 ECMP. A failure of one path redistributes traffic on the remaining paths, resulting in a traffic loss. The availability of a new path results in traffic being redistributed to that path.

VTEP failover

KeepAlive BFD can be configured to monitor the presence of the peer VTEP, in a Layer 3 BFD multihop session with BGP as the client. Alternatively, the BGP connection state can be used as a trigger for any failover actions. When a failover is detected, all the extension and composite cluster clients run a designated forwarder (DF) reelection to elect the surviving VTEP as the DF.

All MACs learnt over the ICL, whether for single– or dual–homed clients, are flushed, and VXLAN transmit and receive BFD takes place on the surviving VTEP.

Tunnel failover

When a tunnel is operationally down, DF reelection procedures are triggered to reelect a DF. KeepAlive BFD is applied in a Layer 3 BFD multihop session with BGP as the client. Alternatively, the BGP connection state can be used as a trigger for any failover actions. When a failover is detected, all the extension and composite cluster clients run a DF reelection to elect the surviving VTEP as the DF.

All MACs learnt over the ICL, whether for single– or dual–homed clients, are flushed, and VXLAN transmit and receive BFD takes place on the surviving VTEP.

ICL failure

Single–homed clients can see persistent packet loss depending on which VTEP the VXLAN termination occurs. Multihomed clients should have the client isolation mode as "loose," to ensure that traffic (BUM or unicast) is always forwarded to the client. Depending on

the traffic flows, it is possible that there is persistent flooding, because learning might take place on one VTEP peer while client-bound traffic is received on the another VTEP peer.

Configuring VXLAN extension tunnels

This task configures a VXLAN overlay gateway to support Layer 2 extension tunnels, with a variety of options.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **overlay-gateway** command and specify an overlay gateway.

```
device(config)# overlay-gateway mygateway
```

3. In VXLAN overlay gateway configuration mode, enter the **type** command and specify **layer2-extension**.

```
device(config-overlay-gateway-mygateway)# type layer2-extension
```

4. Enter the **ip interface** command to specify either loopback or a virtual Ethernet (VE) interface as a Layer 3 interface for a VXLAN overlay gateway.
   - This example specifies a loopback interface.

```
device(config-overlay-gateway-mygateway)# ip interface loopback 10
```

   - This example specifies a VE interface and VRRP-E Virtual Router ID (VRID).

```
device(config-overlay-gateway-mygateway)# ip interface ve 10 vrrp-extended-group 10
```

5. There are two ways to map VLANs to VNIs. You must do one of the following.
   - To enable the automatic mapping of VLANs to VNIs, enter the **map vni auto**

```
device(config-overlay-gateway-mygateway)# map vni auto
```

     NOTE
     The VNI is the VLAN ID (VID) for VIDs, and is 4096 + the bridge domain ID for BDs.

   - To configure the mapping of VLANs to VNIs manually, enter the **map vlan** command. This example using ranging.

```
device(config-overlay-gateway-mygateway)# map vlan 10,20-30 vni 10,20-30
```

6. (Optional) To map a bridge domain (BD) or range of BDs to a VNI or range of VNIs, enter the **map bridge-domain** command. This example using ranging.

```
device(config-overlay-gateway-mygateway)# map bridge-domain 10-30 vni 10-30
```

7. Activate the gateway, by entering the **activate** command.

```
device(config-overlay-gateway-mygateway)# activate
```

8. (Optional) To create a remote Layer 2 extension site in a VXLAN overlay gateway context, enter the **site** command to specify a site and enter VXLAN overlay gateway site configuration mode to configure a variety of options.
   a) Specify a site.

```
device(config-overlay-gateway-mygateway)# site mysite
```

   b) Specify the destination IPv4 address of a tunnel, by using the **ip address** command.

```
device(config-overlay-gateway-mygateway-site-mysite)# ip address 10.11.12.13
```

   c) Configure a switchport VLAN or range of VLANs for the tunnels to the containing site, by using the **extend vlan** command with the **add** keyword. This example uses ranging.

```
device(config-overlay-gateway-mygateway-site-mysite)#  extend vlan add 10,20-30
```

   Use the **remove** keyword to remove the VLANs.

   d) Configure a switchport bridge domain (BD) or range of BDs for the tunnels to the containing site, by using the **extend bridge-domain** command with the **add** keyword. This example uses ranging.

```
device(config-overlay-gateway-mygateway-site-mysite)# extend bridge-domain add 10,20-30
```

   Use the **remove** keyword to remove the BDs.

   e) Enable Bidirectional Forwarding Detection (BFD) for the site.

```
device(config-overlay-gateway-mygateway-site-mysite)# bfd
```

   f) Configure BFD) session parameters for the site, by using the **bfd interval** command.

```
device(config-overlay-gateway-mygateway-site-mysite)# bfd interval 2000 min-rx 3000 multiplier 26
```

   This command can also be used to configure BFD behavior specific to individual interfaces, as appropriate.

   g) Enable the site, by using the **no shutdown** command.

```
device(config-overlay-gateway-mygateway-site-mysite)# no shutdown
```

Configuring logical VTEPs

This section lists the requirements for configuring logical VTEPs (LVTEPs).

Ensure that the following are configured before the overlay gateway is configured:

- BGP EVPN configurations with encapsulation of NSH for the LVTEP peer
- Cluster configuration for the LVTEP peer and ICL interface
- Cluster management with a node ID
- VTEP source IP address identical on both peers

IP-Based Management Cluster

- IP-based management cluster....49
- Configuring an IP-based management cluster....50

IP-based management cluster

IP-based management cluster builds logical clusters of switches over IP, to manage cluster-wide configurations and distribution services.

This feature supports two-node clustering, to support a two-node leaf architecture for IP Fabrics. A node ID is used to identify a switch in the cluster uniquely. By default the node ID is 1. If a node is part of a cluster, an attempt to change the node ID is rejected with an error message. The node ID can be changed only after the node is removed from the cluster.

A node being added to the cluster does not need to be in the default configuration, and a node rejoining the cluster can have different global configuration settings compared to the primary node. Local configuration is preserved for a rejoining node.

At any given time, a node in the management cluster is in one of the following states:

- Standalone (from the node's perspective)
- Primary (from the node's perspective)
- Connected to the cluster (online)
- Disconnected from the cluster (offline)
- Adding in progress (coming online)
- Rejoin in progress

  ATTENTION
  The selection of the principal node may not always be honored. The following are the conditions for a node to become principal.

- If a node is already the principal node, it retains this status.
- If both nodes are standalone/principal nodes, then the node with the nondefault configuration becomes the principal node. The other node in the cluster must be in the default configuration.
- If, for example, Node A is aware of its peer, Node B, and Node B is not aware of Node A, then Node A becomes the principal node. This can happen if the **no peer** command is executed on only a single node in the cluster.
- If none of the above conditions are met—for example, both nodes (1) are standalone nodes, (2) have a nondefault configuration, and (3) are aware of each other—, then the node with the lower node-id becomes the principal node.

The Multi-Chassis Trunk (MCT) **peer** command is used to configure information for both of the peer nodes. The new peer is added to the cluster automatically as long as the peer is reachable through IP connectivity. Once the cluster is formed, any subsequent configuration changes on one node is propagated to the other.

The **no peer** command causes a controlled failover and the target node is removed permanently from the cluster. Primary and secondary controlled failover is supported, as is primary and secondary uncontrolled failover (on loss of heartbeat).

After an entire cluster is rebooted, the **copy default-config startup-config** command takes the entire cluster to the default state, but the node ID is preserved. (The node ID is preserved even if the database becomes corrupted.) The **copy** *source-file* **startup-config** command removes the management cluster configuration.

Virtual IP and IPv6 addresses can also be configured for the cluster.

The following features are not supported for IP-based management cluster:

- Clustering of nodes that have different database schema
- Global configuration from secondary nodes
- Cluster-wide firmware upgrades

Configuring an IP-based management cluster

This task configures a Multi-Chassis Trunk (MCT) peer group and configures an IP-based management cluster.

1. Enter global configuration mode and enter the cluster command to specify the name and cluster ID of an MCT (two-node) cluster.

   ```
   device# configure terminal
   device(config)# cluster MCT1 1
   ```

   The cluster name does not appear in the prompt, but it does appear in the output of the **show cluster** command.

2. In MCT cluster configuration mode, enter the **peer** command to specify an IP address.

   ```
   device(config-cluster-1)# peer 10.10.10.12
   ```

3. (Optional) In global configuration mode, enter the **cluster management virtual** command and specify an IP address and other options as appropriate.

   ```
   device(config)# cluster management virtual ip address 10.10.10.13/24 inband interface ve 4
   ```

4. (Optional) In privileged EXEC mode, enter the **cluster management node-id** command to change the node ID from the default (1).

   ```
   device# cluster management node-id 20
   ```

5. Enter the **show cluster management** command to confirm the configuration.

   ```
   device# show cluster management
   CLUSTER ID                        : 2
   Management Cluster UUID : 455451fd-205f-4482-87d4-4ed55944132c
   Total Number of Nodes in Cluster   : 2

   Node-Id        Switch MAC/WWN               IP Address       Status
   ____________________________________________________________________
   1              10:00:c4:f5:7c:50:06:2e      10.0.0.12        Co-ordinator
   48             10:00:c4:f5:7c:50:06:2d      10.0.0.13        Connected
   ```

Overlay Services

- Overlay services overview....51
- Overlay services provisioning....52
- Overlay policy system maximum numbers....57
- Overlay policy-map inner flow keys and actions supported....58
- Overlay services interaction with hardware profiles....59
- Using overlay policy map commands....60
- Overlay configuration examples....61

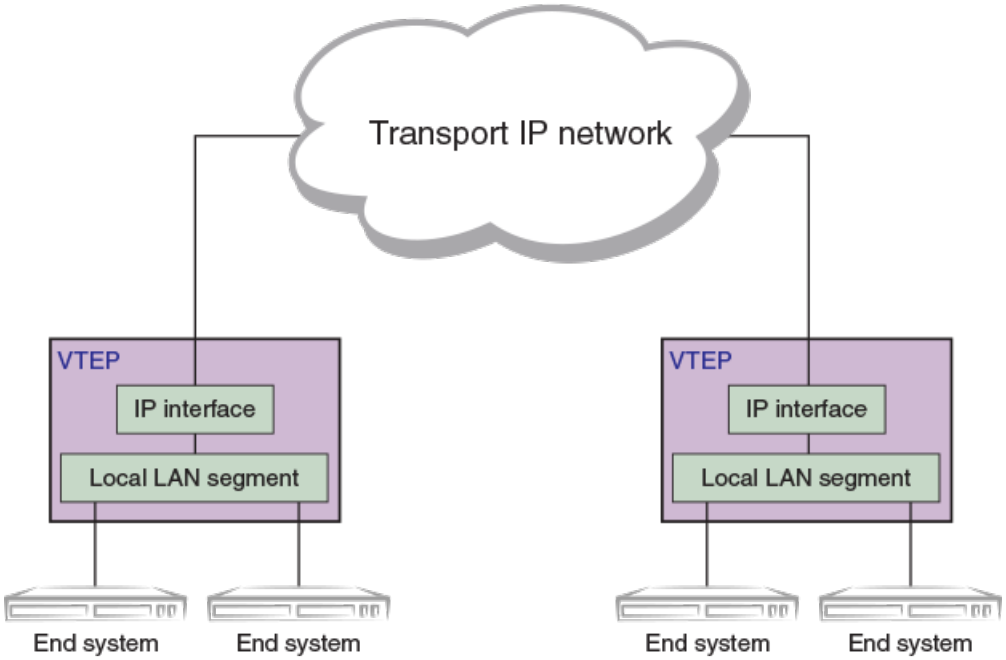
Overlay services overview

A transit service is basically the service of allowing network traffic from one network to cross or "transit" another computer network, usually to connect a smaller Internet service provider to the larger Internet.

This service allows the smaller providers to tunnel Layer 2 packets across a larger Layer 3 network (creating an overlay of the Layer 2 traffic over a Layer 3 network). The tunneling of the Layer 2 packets is achieved by encapsulating the original frame inside a Layer 3 encapsulated tunneling frame such as Virtual eXtensible LAN (VXLAN), Network Virtualization using Generic Routing Encapsulation (NVGRE), or other tunnel-encapsulated packets that typically consist of an outer MAC and outer IP header to identify the tunnel.

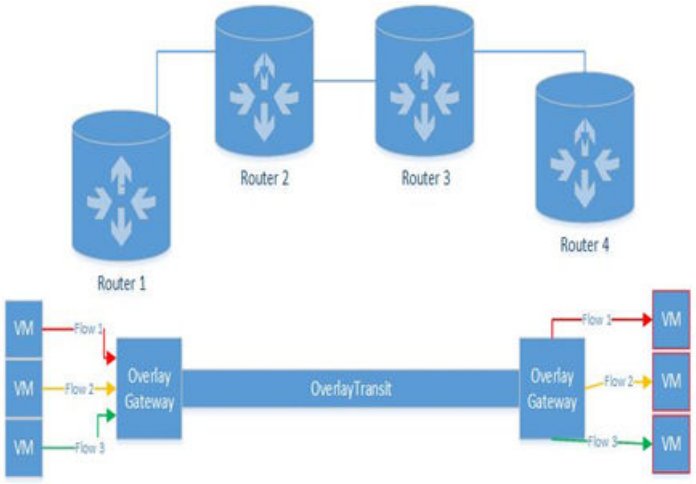
The following figure depicts a VXLAN tunnel. The local LAN segments within each VTEP (VXLAN Transit End Point) connect to the transport IP network through an IP interface. The IP address assigned to the VTEP is the IP address used within the encapsulating outer IP header and identifies one end of the tunnel.

FIGURE 5 VXLAN tunnel



Overlay services operate at two points in the network. The first point is where the tunnel is decapsulated and is referred to as an "overlay gateway service." The other point is within the overlay transport IP network and is referred to as "overlay transit services." In the following figure, an example topology with virtual machine (VM) support, an overlay gateway exists at the tunnel origination and termination points (Router 1 and Router 4). An overlay transit exists on nodes that a tunnel simply traverses (Router 1 and Router 3).

FIGURE 6 Overlay services



It is possible under both points of operation for the "provider" of the services to do access control, policing, SPAN, sFlow, re-marking, and so on by looking into the inner packet of a VXLAN/NVGRE/Other tunnel-encapsulated packet. To perform actions on a flow within a tunneled service, the system must first Identify the tunnel carrying the service (identified by the outer headers), as well as match information carried within the tunneled frames.

While there may be multiple outer encapsulations for tunneling, this discussion focuses on providing flow-based services on traffic being tunneled across VXLAN-encapsulated packets. From an implementation perspective, consideration is given to NVGRE as well.

Overlay services provisioning

This section presents guidelines for creating and applying flow-based services on flows within a tunnel.

The flow-based services are built on top of and reuse the existing ACL infrastructure for inner packet filtering and QoS manipulation. The provisioning hierarchy as used in the control of flows within an overlay transit service, from top to bottom, is shown below:

- Overlay policy
  - Overlay classification
  - Flow-based control

An overlay policy is comprised of a sequence of *stanzas*, each of which contains two components:

- The first component, overlay classification, is a rule or set of rules identifying a tunnel or group of tunnels on which flow-based control is to be performed.
- The second component, flow-based control, is comprised of one or more security ACL bindings, QoS service policy-map applications, or both.

NOTE
Stanzas with lower sequence numbers are considered as a match when overlay classifications overlap.

These components are described further below.

- *Overlay policy*—This object (overlay-policy-map) is the top level object, referred to as the overlay policy map. It contains the information used to identify the transporting tunnel (overlay classification), as well as information to identify individual flows (flow-based control) within the tunnel, if desired, and the resulting action to take on matching flows.
- *Overlay classification*—This object (overlay-class-map) defines the information used to identify the transporting tunnel(s) on which the policy is to apply. A tunnel is identified according to the source or destination IP address, as well as the tunnel type.
- *Flow-based control*—Flow-based control allows the user to perform an action on a specific flow traversing the tunnel. Flow-based control uses the existing ACL and QoS configuration objects (access group and service policies) to identify the desired flow, as well as to specify the desired action to perfom on the flow (on the inner packet).

Overlay policy map rules

The following rules pertaining to class map configuration apply to class maps used in all scenarios.

- An overlay policy map name must begin with a–z, A–Z or 0–9. Underscore and hyphen can also be used, except as the first character, and the name length can not exceed 63 characters.
- The overlay policy must contain an overlay-class-map directive that contains the information necessary to identify the tunnel or tunnels on which the policy is to apply.
- The overlay policy must contain at least one statement for matching the inner flow (achieved through ACL matching) and the action to take on the matching flow. The supported actions are any existing ACL security action (permit or deny), as well as any desired flow-based QoS result (for example, sampling or mirroring).
- The forward referencing of overlay class maps, ACLs, or service-policy maps is not allowed. The user must first define the configuration to be contained in the policy map (overlay class maps, ACLs, or QoS service maps) before referencing them in an overlay policy map. The forward referencing of the configuration causes an error to be returned.

The following sections provide rules specific to either the gateway or transit mode of operation.

Overlay policy map behavior

The following rules govern overlay policy map behavior:

- An actively applied overlay policy map may not be deleted.
- The overlay policy map must exist before it can be applied. If an attempt is made to apply a non-existent policy map, an error will be returned.

Overlay class map rules

The following rules pertaining to class map configuration apply to class maps used in all scenarios:

- An overlay class map name must begin with a–z, A–Z, or 0–9. Underscore and hyphen can also be used, except as the first character, and the name length can not exceed 63 characters.
- An overlay class map can contain up to 1024 rules for identifying tunnels.
- A class map may not be deleted if it is currently referenced by an existing overlay policy map.

- The rules of an overlay class map must match the configured visibility profile. Rules with a key profile that is different from the selected visibility profile will be inactive.

The following sections provide rules specific to either the gateway or transit mode of operation.

Overlay transit services class map rules

The following rule applies to overlay transit services:

- The overlay policy classification is preformed through hash table look-ups (not TCAM tables), and consequently it is entirely possible for the user to enter a valid classification rule even though the rule, even though the rule is not actually programmed as a result of a hash table collision. When this occurs, the user is informed of the collision by means of a RASlog and the classification status is displayed as" unwritten/inactive".

Overlay gateway services class map rules

The following rules apply to overlay gateway services:

- Gateway services take place at the termination of the tunnel and therefore only operate on or pertain to a single tunnel at a time. (For gateway services there is no concept of tunnels or groups of tunnels).
- When a policy map is applied in a "gateway" mode, the class map(s) referred to by the policy map are validated to contain only one rule (or tunnel classification). If more than one class map rule exists at the time of application, the user is informed of the violation by a RASlog and the classification and inner ACLs/QoS policies become inactive.
- The class-map rule can contain only the source IP address or source and destination IP addresses. Other options or key combinations are invalid. The source IP/source + destination IP is used to identify the tunnel interface for termination.

Flow-based classification rules

There is no change in the configuration of security ACLs or QoS policy objects that are used for overlay transit services flow-based classification. However, when an overlay policy is applied, depending on how it is applied (whether for transit or gateway modes), some "key" consistency checks may be performed on the ACL or QoS policy that is specified for classification.

In a gateway mode of operation, there are no underlying constraints on flow-based keys or operation, because the underlying hardware for overlay application is TCAM-based and therefore can support any or all keys and mask settings as security ACL or flow-based QoS.

Only transit service applications are constrained with respect to the flow classification, as described below.

The following helps to clarify the underlying pipeline support for the classification of flows within the tunnel. (More-traditional implementations use TCAM look ups, which are a bit more straightforward.)

For each overly classification that is defined, the transit services subsystem internally defines an allowed "key type pair" for the overlay. The tunnel or groups of tunnels are identified by the outer lookup and the key type pair for the tunnel/tunnel group is passed to a key extraction process. The key extraction process extracts two keys from the frame and passes one key each to the two "inner key hash" processes.

When an overlay policy map is applied, internal logic determines the key types to be used for the classification of the inner flow. The following tables define a key type value and specify which fields are included in the key type. The HASH 1 table is used for the highest priority traffic, so if an incoming frame "matches" a rule within both tables, the result from HASH 1 takes precedence.

TABLE 12 Inner hash table key types: High priority (HASH 1)

ID	VNI	SIP	DIP	DSCP	ULP	L4Info
sip_dip_dscp_ulp_info	*	sip	dip	dscp	ulp	l4info
sip_dip_ulp_info	*	sip	dip	*	ulp	l4info
sip_dip_dscp	*	sip	dip	dscp	*	*

TABLE 12 Inner hash table key types: High priority (HASH 1) (continued)

ID	VNI	SIP	DIP	DSCP	ULP	L4Info
sip_dip_ulp	*	sip	dip	*	ulp	*
sip_ulp_info	*	sip	*	*	ulp	l4info
sip_dscp	*	sip	*	dscp	*	*
dip_ulp_info	*	*	dip	*	ulp	l4info
dip_dscp	*	*	dip	dscp	*	*

TABLE 13 Inner hash table key types: Low priority (HASH 2)

ID	VNI	SIP	DIP	DSCP	ULP	L4Info (see Note)
sip_dip_dscp_ulp	*	sip	dip	dscp	ulp	*
sip_dip_ulp	*	sip	dip	*	ulp	*
sip_dip	*	sip	dip	*	*	*
sip_dip	*	sip	dip	*	*	*
sip_ulp	*	sip	*	*	ulp	*
sip	*	sip	*	*	*	*
dip_ulp	*	*	dip	*	ulp	*
dip	*	*	dip	*	*	*

NOTE

"L4Info" is defined as source or destination port. If the source port is specified, it must have a value that is less than 1024.

To determine the key types, the logic examines the keys specified in the first rule of the ACL and a key ID from either table is assigned to the ACL and is provided as a "result" field to the outer key HASH 1 lookup. The logic continues to examine and instantiate all rules of the ACL. Upon encountering a subsequent rule that cannot use the same key ID as that assigned to the first rule, the logic maps the rule to a key ID from the remaining unused inner key hash table. For example, if the first rule mapped to a rule within HASH 1, then upon detection of a nonmatching rule type a key is identified and allocated from HASH 2. If any or all remaining rules cannot use either of the two key types allocated for HASH 1 or HASH 2, the rules are not programmed and are displayed as "inactive".

The following rules govern flow-based classification behavior:

- All rules defined in the flow classification ACL must contain only rules that can map onto one key type from each of the two hash table definitions above.
- The first rule entered within the ACL defines the key fields to be used from a particular table for the associated overlay classification. If the second rule within the ACL does not use the same keys, a key type is allocated from the "other" table. (That is, if the first rule defines a key within the HASH 1 table, then a rule is defined for the HASH 2 table and vice versa.) Once a key type has been assigned to each hash table (as determined by the ACL rule content), all subsequent rules of the ACL must use the same key type assigned to one of the two tables in order to be instantiated in the hardware.
- Rule validation for key consistency is performed when the ACL is instantiated as a result of an overlay policy binding. Any rules not following the two key types are not programmed. The resulting rule shows as "unwritten" and a RASlog indicating a key type mismatch is logged at the time of binding.
- While flow-based classification is based on user ACL configuration (that is, TCAM based), the flow-based classification for transit services is performed by means of a hash lookup. Using a hash table for lookup means that it is entirely possible for the user to enter a valid flow classification rule (ACL rule), even though the rule, when programmed, is not programmed because of a hash table collision. When this occurs the user is informed of the collision by a RASlog, and the rule status is displayed as " unwritten".

- Only "equal to" operator is allowed for Layer 4 ports.
- Only the source or the destination port must be present in the rule. Both source and destination port is not allowed. Such rules are errored out as "invalid keytype".
- If the source port is specified it must have a value that is less than 1024.
- A given packet source port is used in L4Info for lookup if the source port number is less than 1024 and the destination port number is greater than 1024. Otherwise, the destination port value is provided to L4Info for lookup.

Flow-based classification behavior

Flow-based classification uses the existing ACL and QoS configuration objects. This section defines behaviors that are consistent between the two modes of application. Any differences based on the mode of operation are addressed below.

The following general rule governs flow-based classification behavior:

- The forward referencing of ACLs or service-policy maps is not allowed. The user must first define the objects to be contained in the policy map (ACLs or Qos service maps) before referencing them in an overlay policy map. The forward referencing of an object is "errored" during the configuration.

Transit services flow-based classification behavior

The following rules apply to transit services:

- For all the ACLs for a given overlay class map, the rules must map into one of two sets of key types for the rule. In other words, for a given overlay class map that contains ACLs A, B, and C, then all the rules of A, B, and C must contain the same set of keys to match, and the set of keys must be map into one of two sets defined for the overlay class map.
- For each overlay classification two sets of flow-based classification keys are possible. The two sets are not fixed and depend on the contents of the specified ACL. When the ACL is applied, the transit services logic defines a set based on the contents of the ACL. Once a subsequent rule is encountered that does not map to the same keys as defined in the first rule, a "second" key type is defined by the logic. From here on all remaining rules must map to one of the two detected key sets in order to be instantiated into the hardware.
- ACL rules must contain only exact matches (that is, masks, ranges, and so on are not supported).
- Layer 4 source port, TCP flags, and VLAN keys are not supported for matching.
- MAC and IPV6 ACLs are not supported.
- Counters for ACLs in a service policy for a given ACL rule are supported when the same rule does not exist in the security ACL.

Overlay policy map inner flow keys and actions supported

This section provides information pertaining to which features of flow-based QoS and security ACLs are applicable to overlay policies. While overlay policies use the existing security ACL and service policy constructs, not all the capabilities of those features are supported. This section details the supported ACLs, ACL keys, and key types.

The following table lists support for ACL matching.

TABLE 14 Support for ACL matching

MAC	IPv4	IPv6
No	Yes	No

The following table lists support for IPv4 ACL keys.

TABLE 15 Support for IPv4 ACL keys

	Src IP	Dst IP	IP Proto	DSCP	L4 Src Port	L4 Dst Port	TCP Flags	DSCP
	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Masks	No	No	No	No	No	No	No	No

The following table lists the QoS services that are supported for flows within a tunnel. All unsupported QoS actions are ignored when the service policy is applied as part of an overlay policy map. A RASlog captures the fact that the specified configuration is not fully instantiated.

TABLE 16 Support for flow-based QoS

Span	sFlow	Policing	DSCP Remarking	COS Remarking	Count	Log
Yes	Yes	No	No	No	No	No

NOTE
sFlow and Count are mutually exclusive. When both are specified only the sFlow action is performed.

The following table depicts the secondary actions support by a user ACL applied within an overlay map.

TABLE 17 Support for secondary actions

Count	Log
Yes	No

Overlay gateway services flow-based classification behavior

The following rules apply to overlay gateway services:

- All keys, masks, and results supported by ACL and flow-based QoS are supported.
- MAC IPV4 and IPV6 ACLs are supported.
- If the tunnel interface identified by the overlay class map does not exist, then the inner ACL and QoS service policy becomes inactive.
- The inner ACLs and QoS policies that are active for an existing tunnel also depend on the TCAM profile that is currently set.

Overlay policy system maximum numbers

The following table lists the system maximum numbers for overlay policies.

TABLE 18 Overlay policy system maximums

System Resource	Maximums
Number of configured overlay policies	256
Number of actively applied overlay policies	1
Number of stanzas per overlay policy	256
Number of overlay classification rules per class-map	1024
Number of flow-based classification rules	512

NOTE
For additional information, see "ACL and rule limits" in the *Extreme SLX-OS Security Configuration Guide*.

Overlay policy-map inner flow keys and actions supported

This section provides information pertaining to which features of flow-based QoS and security ACLs are applicable to overlay policies.

While overlay policies use the existing security ACL and service policy constructs, not all capabilities of those features are supported.

The supported ACLs and ACL keys and key types are detailed below.

ACL matches supported

The following table lists the ACL matches that are supported.

TABLE 19 ACL matches supported

MAC	IPv4	IPv6
No	Yes	No

IPv4 ACL keys supported

The following table lists the IPv4 ACL keys that are supported.

TABLE 20 IPv4 ACL keys supported

	Src IP	Dst IP	IP Proto	DSCP	L4 Src Port	L4 Dst Port	TCP Flags	DSCP	
	Yes	Yes	Yes	Yes	No	Yes	No	Yes	
Masks	No	No	No	No	No	No	No	No	

Flow-based QoS

The following table depicts the QoS services that are supported for flows within a tunnel. All unsupported QoS actions are ignored when the service policy is applied as part of an overlay policy map. A RASlog entry captures the fact that a specified configuration is not fully instantiated.

TABLE 21 QoS services supported

Span	sFlow	Policing	DSCP re-marking	COS re-marking	Count	Log		
Yes	Yes	No	No	No	Yes	No		

NOTE
sFlow and Count are mutually exclusive. When both are specified, only the sFlow action is performed.

Security ACLs

The following table depicts the security actions supported by a user ACL that is applied within an overlay map.

TABLE 22 Security ACL actions

Count	Log
Yes	No

NOTE

"Count" is a secondary action. Security ACLs must all contain a primary action of "permit" or "deny," and every rule must have a primary action. Secondary actions are optional.

Overlay services interaction with hardware profiles

Hardware profiles must be configured to support the appropriate classification method.

Overlay visibility profiles

The overlay visibility profile must be set to the appropriate classification method. If the class map rules do not match the visibility profile selected, the classification is not programmed into the hardware when applied within the context of an overlay transit service. This profile is configured by means of the **profile overlay-visibility** command in hardware configuration mode, with options as shown below.

Overlay visibility profile options

Enter the **profile overlay-visibility** command in hardware configuration mode to view available options.

```
device# configure terminal
device(config)#
device(config-hardware)# profile overlay-visibility ?
Possible completions:
  default        match on outer source IP and destination IP
  endpoint       match on outer source IP or destination IP
  endpoint-vni   match on outer source IP and VNI or destination IP and VNI
  vni            match on VNI only
```

Overlay gateway bindings and TCAM profiles

When applying overlay policies to an overlay gateway, you must ensure that the appropriate TCAM profile is selected to support the desired security or QoS flow classification. If the TCAM profile does not support the specified flow classification, a RASlog is generated and the ACL programming status is shown as “inactive”.

TCAM profile options vary by platform, as shown below.

TCAM profile options for the SLX 9140

Enter the **profile tcam ?** command in hardware configuration mode to view available options for this platform.

```
SLX9140(config)# hardware
SLX9140(config-hardware)# profile tcam ?
Possible completions:
  default              basic support for all applications
  l2-acl-l3-iacl       ingress, egress L2 ACL & ingress IPv4, IPv6 ACL
  l2-iacl-l3-acl       ingress L2 ACL & ingress, egress IPv4, IPv6 ACL
  l2-l3-iacl-l2-iqos   ingress L2, IPv4, IPv6 ACL & ingress L2 QoS
  l2-l3-iqos-l2-eacl   ingress L2, IPv4, IPv6 Qos & egress L2 ACL
  l2-l3-iqos-l2-iacl   ingress L2, IPv4, IPv6 Qos & ingress L2 ACL
  l2-l3-iqos-l3-eacl   ingress L2, IPv4, IPv6 Qos & egress IPv4, IPv6 ACL
  l2-l3-iqos-l3-iacl   ingress L2, IPv4, IPv6 Qos & ingress IPv4, IPv6 ACL
  user-defined         user-defined profile
```

TCAM profile options for the SLX 9240

Enter the **profile tcam ?** command in hardware configuration mode to view available options for this platform.

```
SLX9240(config)# hardware
SLX9240(config-hardware)# profile tcam ?
Possible completions:
  default           basic support for all applications
  l2-l3-iacl        ingress L2 & ingress IPv4, IPv6 ACL
  l2-l3-iqos        ingress L2, IPv4, IPv6 Qos
  l3-acl            ingress, egress IPv4, IPv6 ACL
  l3-iacl-l2-eacl   ingress IPv4, IPv6 ACL & egress L2 ACL
  l3-iacl-l2-iqos   ingress IPv4, IPv6 ACL & ingress L2 Qos
  l3-iqos-l2-iacl   ingress IPv4, IPv6 Qos & ingress L2 ACL
  user-defined      user-defined profile
```

Using overlay policy map commands

Overlay policy maps are the top-level control objects for transit services.

The user creates the overlay policy map by defining the set of tunnels or VTEPs to which the policy map is to apply. The tunnel/VTEP definition is provided by the filters created with by means of the **overlay-class-map** command. Finally, the user specifies the "flow" to match within the tunnel/VTEPs by specifying an IP access group or QoS service policy thata contains a match on IPv4 ACLs. The action to take is derived from either the security policy (permit/deny) or the service policy (SPAN/sFlow).

An overlay policy map can consist one or more "stanzas," where a stanza is a group of directives consisting of one classification statement and one or more "flow" actions. The sequence for creating a policy map is as follows:

1. Create the policy map container.
2. Specify the stanza number (sequence number) and define the tunnel classification to use (assign a class map).
3. Specify the flows within the classified tunnel to which to apply specific actions.
4. Repeat steps 2 and 3 for additional stanzas.

Overlay configuration examples

This section presents a variety of overlay configuration examples, as well as the outputs of related **show** commands.

> NOTE
> For additional details, refer to the *Extreme SLX-OS Security Configuration Guide*.

The user should first configure any flow-based ACLs or service policies that will be referenced by the overlay policy map, as shown in the following example.

```
ip access-list extended ipHostFilter
 seq 10 permit ip host 171.171.171.171 host 174.174.174.174 count
 seq 20 permit ip host 1.1.1.1 host 2.2.2.2 count
 seq 30 permit tcp any host 55.55.55.55 count
!
ip access-list extended tcpHostDscpFilter
 seq 10 permit tcp host 9.8.7.6 host 5.4.3.2 dscp 10 count
 seq 20 permit tcp host 11.11.11.11 host 22.22.22.22 dscp 20 count
 seq 30 permit tcp host 55.55.55.55 host 66.66.66.66 dscp 30 count
!
sflow-profile mySflow sampling-rate 1024
sflow-profile anotherSample sampling-rate 512
!
class-map qosClass
 match access-group tcpHostDscpFilter
!
policy-map qosMap
 class qosClass
 sflow-profile mySflow
 exit
exit
!
```

All rules contained within the access list that will be used to classify the flow within the overlay are required to contain all of the same key types. Two sets of key definitions are assigned to an ACL used in overlay flow classifications. All rules contained in the ACL must map to one of the two sets assigned to the overlay classification.

Next the user needs to configure the classification criteria for all of the overlays on which they would like to perform some type of flow action. The overlay classification is achieved through an overlay class map, as shown in the following example.

```
overlay-class-map tunnel-group-1
 seq 10 match source 2.2.2.1 destination 10.1.1.2
 seq 20 match source 8.8.8.8 destination 20.1.1.2

overlay-class-map tunnel-group-2
 seq 10 match source 12.22.22.1 destination 100.1.1.2
 seq 20 match source 18.81.58.8 destination 200.1.1.2
```

All overlay class map rules must specify a single filtering type (source, destination, endpoint, and so on). The platform does not allow mixing of types within the rules.

Once the overlay classifications and flow classification and actions have been defined, the final step is to configure the policy map that associates the overlay and inner flows. This is accomplished by configuring the overlay policy map, then applying the map to a node supporting the transit services container, as in the following example.

```
overlay-policy-map myMap
  seq 10 overlay-class tunnel-group-1
    ip access-group ipHostFilter
  seq 20 overlay-class tunnel-group-2
    ip access-group ipHostFilter
    service-policy qosMap
!
```

```
overlay-transit transit1
 overlay-service-policy in myMap
```

Use the **show overlay-service-policy** command to display overlay transit and ingress policies, as in the following example.

```
device# show overlay-service-policy
Overlay-transit BASIC_VERIFICATION
Ingress policy BASIC_KEYTYPE_VERIFY
```

Use the **show overlay-service-policy** command and specify a service policy for a high-level view of the policy, as in the following example.

```
device# show overlay-service-policy BASIC_KEYTYPE_VERIFY
overlay-policy-map BASIC_KEYTYPE_VERIFY
  Overlay-transit BASIC_VERIFICATION at Ingress
    seq 10 overlay-class OVER_CLASS_1 (Active)
      ip access-group IPv4ACL_High_KT0 (Active)
    seq 20 overlay-class OVER_CLASS_2 (Active)
      ip access-group IPv4ACL_High_KT1 (Active)
    seq 30 overlay-class OVER_CLASS_3 (Active)
      ip access-group IPv4ACL_High_KT2 (Active)
    seq 40 overlay-class OVER_CLASS_4 (Active)
      ip access-group IPv4ACL_High_KT3 (Active)
```

Use the **show overlay-service-policy** command with the **detail** keyword for additional information, as in the following example.

```
device# show overlay-service-policy BASIC_KEYTYPE_VERIFY detail
overlay-policy-map BASIC_KEYTYPE_VERIFY
  Overlay-transit BASIC_VERIFICATION at Ingress
    seq 10 overlay-class OVER_CLASS_1 (Active)
        seq 1 match source 10.1.0.1 destination 20.1.0.1 encap-type vxlan (Active)
        seq 2 match source 10.1.0.2 destination 20.1.0.2 encap-type vxlan (Active)
        seq 3 match source 10.1.0.3 destination 20.1.0.3 encap-type vxlan (Active)
        seq 4 match source 10.1.0.4 destination 20.1.0.4 encap-type vxlan (Active)
      ip access-group IPv4ACL_High_KT0 (Active)
    seq 20 overlay-class OVER_CLASS_2 (Active)
        seq 1 match source 10.2.0.1 destination 20.2.0.1 encap-type vxlan (Active)
        seq 2 match source 10.2.0.2 destination 20.2.0.2 encap-type vxlan (Active)
        seq 3 match source 10.2.0.3 destination 20.2.0.3 encap-type vxlan (Active)
        seq 4 match source 10.2.0.4 destination 20.2.0.4 encap-type vxlan (Active)
      ip access-group IPv4ACL_High_KT1 (Active)
    seq 30 overlay-class OVER_CLASS_3 (Active)
        seq 1 match source 10.3.0.1 destination 20.3.0.1 encap-type vxlan (Active)
        seq 2 match source 10.3.0.2 destination 20.3.0.2 encap-type vxlan (Active)
        seq 3 match source 10.3.0.3 destination 20.3.0.3 encap-type vxlan (Active)
        seq 4 match source 10.3.0.4 destination 20.3.0.4 encap-type vxlan (Active)
      ip access-group IPv4ACL_High_KT2 (Active)
    seq 40 overlay-class OVER_CLASS_4 (Active)
        seq 1 match source 10.4.0.1 destination 20.4.0.1 encap-type vxlan (Active)
        seq 2 match source 10.4.0.2 destination 20.4.0.2 encap-type vxlan (Active)
        seq 3 match source 10.4.0.3 destination 20.4.0.3 encap-type vxlan (Active)
        seq 4 match source 10.4.0.4 destination 20.4.0.4 encap-type vxlan (Active)
      ip access-group IPv4ACL_High_KT3 (Active)
```

Use the **show overlay-policy-map** command to display all information, as in the following example.

```
device# show overlay-policy-map
overlay-policy-map BASIC_KEYTYPE_VERIFY
  Configuration:
    seq 10 overlay-class OVER_CLASS_1
      ip access-group IPv4ACL_High_KT0
    seq 20 overlay-class OVER_CLASS_2
      ip access-group IPv4ACL_High_KT1
    seq 30 overlay-class OVER_CLASS_3
      ip access-group IPv4ACL_High_KT2
    seq 40 overlay-class OVER_CLASS_4
      ip access-group IPv4ACL_High_KT3
  Active On:
    Overlay-transit BASIC_VERIFICATION
```

Logical Interfaces

- Logical interfaces overview....63
- Configuring a logical interface on a physical port or port-channel (LAG)....65

Logical interfaces overview

A logical interface (LIF) serves the purpose of abstracting a forwarding interface in a very generic way, making it possible to capture the underlying physical characteristics of a forwarding interface.

This feature facilitates the support of future forwarding technologies without the need to modify code design in various software components.

A forwarding interface is also known as "main interface." It can be a physical port, a port-channel (Link Aggregation Group, or LAG), a pseudowire (PW), a tunnel, and so on. A logical interface can also be thought of as a subinterface configuration on top of a main interface.

NOTE
Currently the only LIFs that require explicit user configuration are attachment circuit (AC) LIFs.

LIFs and bridge domains

A Layer 2 application for LIFs is for bridge domains (BDs). A BD is an infrastructure that supports the implementation of different switching technologies; it is essentially a generic broadcast/forwarding domain that is not tied to a specific transport technology. Bridge domains support a wide range of service endpoints, including regular Layer 2 endpoints and Layer 2 endpoints over Layer 3 technologies. Logical interfaces representing BD endpoints must be created before they can be bound to a BD. For more information and configuration details, refer to the Bridge Domains on page 67 chapter in this guide.

Configuration considerations

The following are some common rules to consider in configuring logical interfaces:

- By default, when the LIF is created it is configured as "no shutdown."
- By default, when the LIF is created, it is "tagged" unless it is explicitly configured with the "untagged" option.
- Allowed LIF service instance ID ranges are from 1 through 12288.
- An LIF service instance ID has no correlation to the VLAN ID of the LIF.
- Each physical/LAG-based LIF must have an associated VLAN configured or else it will not be usable when the user attempts to add it to a service (such as VPLS, Layer 2). Such a configuration request to add the LIF to a service will be rejected.
- Once the LIF is associated with a Layer 2 service, its VLAN value cannot be changed or deleted unless it is first removed from the associated service. In case the LIF is not yet associated to a service, the user is free to remove the VLAN configuration or change the VLAN assignment.
- The **no** option to the **logical-interface** command can be applied at any time.
- The "untagged" configuration is allowed for only one LIF under the same physical port or LAG. If one LIF is already configured as untagged, all subsequent attempts on the same physical port or LAG are rejected.
- Once the "untagged" option is selected, it will only have one VLAN as the next classification option. There is no dual-tag support for the untagged case.

- In order to configure an untagged LIF, the main interface must be configured as "switchport mode trunk-no-default-native". If it is configured set to regular trunk mode, the native VLAN is already associated with a regular Layer 2 VLAN LIF and no explicit untagged LIF can be configured on that interface.
- Once the LIF is associated with a service (Layer 2) such as a bridge domain, its "untagged/tagged" configuration cannot be changed. The service instance or its current VLAN classification must be deleted by the user first and then added back with the proper "untagged/tagged" option.
- VLANs 4091 through 4095 are reserved VLANs and these should not be used as the VLAN ID for either the inner or outer VLAN of the LIF.
- The VLAN specified under the LIF ensures that such a VLAN is not already configured under the **switchport** command for a regular Layer 2 allowed VLAN.

If the interface is already configured as "switchport access," then it is not allowed to be configured with LIF. The reverse condition is also not allowed: the interface cannot be changed to mode access if a LIF is still configured under the main interface.

Configuring a logical interface on a physical port or port-channel (LAG)

This task configures a logical interface on a physical (Ethernet) port and a port-channel (Link Aggregation Group, or LAG) interface.

Refer to the Usage Guidelines for the **logical-interface** command for complete details.

1. Do the following to configure a logical interface on an Ethernet port.
   a) Enter global configuration mode.

      ```
      device# configure terminal
      ```

   b) Specify an Ethernet interface.

      ```
      device(config)# interface ethernet 0/6
      ```

   c) Enter the **switchport** command to configure the parent interface as switchport.

      ```
      device(conf-if-eth-0/6)# switchport
      ```

   d) Enter the **switchport mode trunk-no-default-native** command to enable an explicit untagged LIF to be configured.

      ```
      device(conf-if-eth-0/6)# switchport mode trunk-no-default-native
      ```

   e) Enable the interface.

      ```
      device(conf-if-eth-0/6)# no shutdown
      ```

   f) Enter the **logical-interface** command, specify a service instance, and enter LIF configuration mode.

      ```
      device(conf-if-eth-0/6)# logical-interface ethernet 2/6.120
      ```

   g) (Optional) Enter the **name** command to facilitate the management of the LIF.

      ```
      device(conf-if-eth-lif-0/6.120)# name myLIF120
      ```

   h) Enter the **vlan** command with the **inner-vlan** option to specify an interface and create dual-tag VLANs.

      ```
      device(conf-if-eth-lif-0/6.120)# vlan 120 inner-vlan 200
      ```

   i) Alternatively, enter the **untagged vlan** command to specify that the LIF is to receive untagged packets.

      ```
      device(conf-if-eth-lif-0/6.120)# untagged vlan 120
      ```

      See the Usage Guidelines for the **vlan (LIF)** command.

   j) (Optional) By default, the administrative state of the LIF is "no shutdown." To remove the port from participating in any data traffic without having to shut down the physical interface, enter the **no** form of the **shutdown (LIF)** command.

      ```
      device(conf-if-eth-lif-0/6.120)# no shutdown
      ```

   k) (Optional) For convenience, you can also enter up to two options in a single command line, as in the following examples.

      ```
      device(conf-if-eth-0/6)# logical-interface ethernet 0/6.120 name myLIF120
      device(conf-if-eth-0/6)# logical-interface ethernet 0/6.120 vlan 120
      ```

2. To configure a port-channel, configure the basic LIF parameters and options as in Step 1.
   a) Specify a port-channel, set its mode to "trunk-no-default-native," and specify a logical interface service instance.

```
device(config)# interface port-channel 10
device(config-port-channel-10)# switchport mode trunk-no-default-native
device(config-port-channel-10)# logical-interface port-channel 10.3
device(config-if-po-lif-10.3)#
```

   b) Repeat additional substeps in Step 1 as appropriate.

Bridge Domains

- Bridge domain overview....67
- Configuring a bridge domain....67
- Displaying bridge-domain configuration information....68
- Enabling statistics on a bridge domain....71
- Displaying statistics for logical interfaces in bridge domains....72
- Clearing statistics on bridge domains....73

Bridge domain overview

Bridge domain is an infrastructure that supports the implementation of different switching technologies.

A bridge domain is a generic broadcast domain that is not tied to a specific transport technology. Bridge domains support a wide range of service endpoints including regular L2 endpoints and L2 endpoints over L3 technologies.

The following are examples of bridge-domain capable services:

- VPLS—with multiple AC endpoints and pseudowire (PW) logical interfaces (LIFs)
- Local VPLS—with multiple AC endpoints
- VLL—with one AC endpoint and one PW endpoint
- VLL—with two AC endpoints

A bridge domain that is created for a VPLS application is also referred to as a VPLS instance.

Bridge domain statistics

Devices gather statistics for all the logical interfaces and peers in bridge domains.

Statistics must be manually enabled for a specific bridge domain, since statistics for bridge domains are not enabled by default.

Use the **statistics** command in bridge domain configuration mode to enable statistics on a bridge domain.

NOTE

- The statistics reported are not real-time statistics since they depend upon the load on the system.
- Enabling statistics on a bridge domain has a heavy impact on data traffic.
- For optimum utilization of the statistics resources in the hardware, statistics on a bridge domain are not enabled by default.

Configuring a bridge domain

Bridge domains are configured independently of the different switching technologies that they support.

Before configuring a bridge domain, configure any logical interface that is to be bound to the bridge domain. Logical interfaces that represent bridge-domain endpoints must be created before they are bound to a bridge domain. For further information on configuration of logical interfaces, refer to *Logical Interfaces*.

There is an example at the end of this task that shows all the configuration steps in order.

Perform the following task to configure a bridge domain.

1. From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Create a bridge domain.

```
device(config)# bridge-domain 5 p2p
```

   By default, the bridge-domain service type is multipoint (**p2mp**). In this example, bridge domain 5 is configured as a point-to-point service (**p2p**).

3. NOTE
   Logical interfaces representing bridge-domain endpoints must be created before they can be bound to a bridge domain. For further information, refer to *Logical Interfaces*.

   Bind the logical interfaces for attachment circuit endpoints to the bridge domain.

```
device(config-bridge-domain-5)# logical-interface ethernet 0/6.400
```

   In this example, Ethernet logical interface 0/6.400 is bound to bridge domain 5.

4. Repeat Step 4 to bind other logical interfaces for attachment circuit endpoints to the bridge domain.

```
device(config-bridge-domain-5)# logical-interface port-channel 2.200
```

   In this example, port channel logical interface 2.200 is bound to bridge domain 5.

5. (Optional) Enable local switching for bridge domain 5.

```
device(config-bridge-domain-5)# local-switching
```

   By default, local switching is enabled.

6. (Optional) Enable dropping L2 bridge protocol data units (BPDUs) for bridge domain 5.

```
device(config-bridge-domain-5)# bpdu-drop-enable
```

The following example creates bridge domain 5. It binds ethernet and port-channel logical interfaces to the bridge domain. It configures local switching, and enables dropping of L2 BPDUs.

```
device# configure terminal
device(config)# bridge-domain 5
device(config-bridge-domain-5)# logical-interface ethernet 0/6.400
device(config-bridge-domain-5)# logical-interface port-channel 2.200
device(config-bridge-domain-5)# local-switching
device(config-bridge-domain-5)# bpdu-drop-enable
```

Displaying bridge-domain configuration information

Various show commands can be used to display bridge-domain configuration information.

- Enter the **show bridge-domain** command to display information about all configured bridge domains.

```
device# show bridge-domain

Total Number of bridge-domains: 3
Number of bridge-domains: 3

Bridge-domain 1
------------------------------
Bridge-domain Type: mp , VC-ID: 5
Number of configured end-points:  5 , Number of Active end-points: 4
```

```
VE if-indx: 1207959555, Local switching: TRUE, bpdu-drop-enable:TRUE
PW-profile: 1, mac-limit: 128000
Number of Mac's learned:90000,     Static-mac count: 10,
VLAN: 100, Tagged ports: 2(2 up), Un-tagged ports: 0 (0 up)
Tagged ports: Eth 0/6, eth 0/8
Un-tagged ports:

Total PW peers: 2 (2 Operational)
Peer address: 12.12.12.12, State: Operational, Uptime: 2 hr 55 min
      Load-balance: True , Cos enabled:False,
      Assigned LSP;s:
      Tnnl in use: tnl2[RSVP]
      Local VC lbl: 983040, Remote VC lbl: 983040
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 15.15.15.15, State: Operational, Uptime: 2 hr 55 min
         Load-balance: False , Cos enabled:False,
      Assigned LSP's: lsp1, lsp2
      Tnnl in use: tnl1[MPLS]
      Local VC lbl: 983041, Remote VC lbl: 983043
      Local VC MTU: 1500, Remote VC MTU: 1500 ,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)

Bridge-domain 2
-------------------------------
Bridge-domain Type: mp , VC-ID: 100
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx:  NA, Local switching: FALSE, bpdu-drop-enable:FALSE
PW-profile: profile_1, mac-limit: 262144
Number of Mac's learned:90000,     Static-mac count: 10,
VLAN: 100, Tagged ports: 2(1 up), Un-tagged ports: 0 (0 up)
      Tagged ports: eth 0/10, eth 0/10
      Un-tagged ports:
VLAN: 150, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
      Tagged ports: eth 0/5
      Un-tagged ports:

Bridge-domain 3
-------------------------------
Bridge-domain Type: mp , VC-ID: 200
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx: 120793855, Local switching: FALSE, bpdu-drop-enable:FALSE
PW-profile: 2, mac-limit: 262144
Number of Mac's learned:90000,     Static-mac count: 10,
Local switching: TRUE,
VLAN: 500, Tagged ports: 2(2 up), Un-tagged ports: 2 (1 up)
Tagged ports:     eth 0/6, eth 0/3
Un-tagged ports:

Total VPLS peers: 3 (2 Operational)
Peer address: 5.5.5.5, State: Operational, Uptime: 2 hr 35 min
         Load-balance: False , Cos enabled:False,
      Assigned LSP;s:
      Tnnl in use: tnl2[RSVP]
      Local VC lbl: 983050, Remote VC lbl: 983050
      Local VC MTU: 1500,Remote VC MTU: 1500,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 20.20.20.20, State: Operational, Uptime: 0 hr 18 min
         Load-balance: False , Cos enabled:True,
      Assigned LSP's:
      Tnnl in use: NA,
      Local VC lbl: NA, Remote VC lbl: NA
      Local VC MTU: 1500,Remote VC MTU: 1500,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 10.10.10.10, State: Not-Operational (Tunnel Not Available),
         Load-balance: True , Cos enabled:False,
      Assigned LSP's: lsp10, lsp15
      Tnnl in use: NA,
      Peer Index:2
      Local VC lbl: NA, Remote VC lbl: NA
      Local VC MTU: 1500,Remote VC MTU: NA ,
      Local VC-Type: Ethernet(0x05), Remote VC-Type: NA
```

- Enter the **show bridge-domain** command specifying the bridge-domain ID to display information about a specific bridge domain. The following example displays information about bridge domain 501.

```
device# show bridge-domain 501

Bridge-domain 501
------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth 0/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 10.9.9.9, State: Operational, uptime: 2 sec
      Load-balance: False, Cos Enabled: False,
      Tunnel cnt: 1
      rsvp  p101(cos_enable:False cos_value:0)
      Assigned LSPs count:0 Assigned LSPs:
      Local VC lbl: 989042, Remote VC lbl: 983040,
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: 5, Remote VC-Type: 5
```

The following example shows information about a bridge domain (501) in which the **load-balance** option is configured for the peer device 10.9.9.9.

```
show bridge-domain 501

Bridge-domain 501
------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth 0/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 10.9.9.9, State: Operational, uptime: 48 sec
      Load-balance: True , Cos Enabled: False,
      Tunnel cnt: 16
      rsvp  p101(cos_enable:False cos_value:0)
      rsvp  p102(cos_enable:False cos_value:0)
      rsvp  p103(cos_enable:False cos_value:0)
      rsvp  p104(cos_enable:False cos_value:0)
      rsvp  p105(cos_enable:False cos_value:0)
      rsvp  p106(cos_enable:False cos_value:0)
      rsvp  p107(cos_enable:False cos_value:0)
      rsvp  p108(cos_enable:False cos_value:0)
      rsvp  p109(cos_enable:False cos_value:0)
      rsvp  p110(cos_enable:False cos_value:0)
      rsvp  p111(cos_enable:False cos_value:0)
      rsvp  p112(cos_enable:False cos_value:0)
      rsvp  p113(cos_enable:False cos_value:0)
      rsvp  p114(cos_enable:False cos_value:0)
      rsvp  p115(cos_enable:False cos_value:0)
      rsvp  p116(cos_enable:False cos_value:0)
      Assigned LSPs count:0 Assigned LSPs:
      Local VC lbl: 989040, Remote VC lbl: 983040,
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: 5, Remote VC-Type: 5
```

The following example shows information about bridge domain 501 in which the **load-balance** option and four assigned label-switched paths (p101, p102, p103, and p104) are configured for the peer device 10.9.9.9.

```
device# show bridge-domain 501

Bridge-domain 501
------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth 0/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 10.9.9.9, State: Operational, uptime: 4 sec
      Load-balance: True , Cos Enabled: False,
      Tunnel cnt: 4
      rsvp  p101(cos_enable:False cos_value:0)
      rsvp  p102(cos_enable:False cos_value:0)
      rsvp  p103(cos_enable:False cos_value:0)
      rsvp  p104(cos_enable:False cos_value:0)
      Assigned LSPs count:4 Assigned LSPs:p101 p102 p103 p104
      Local VC lbl: 989041, Remote VC lbl: 983040,
      Local VC MTU: 1500, Remote VC MTU: 1500,
      Local VC-Type: 5, Remote VC-Type: 5
```

- Enter the **show bridge-domain brief** command to display summary information about all configured bridge domains.

```
device# show bridge-domain brief

Total Number of bridge-domains configured: 10
Number of VPLS bridge-domains: 5
Macs Dynamically learned: 50360, Macs statically configured: 0

BDID(VC-ID)   TYPE     Intf(up)     PWs(up)  macs
501(501)      P2MP     5(3)         2(2)     50000
502(502)      P2MP     1(1)         1(1)     10
503(503)      P2MP     10(6)        3(1)     0
504(504)      P2MP     1(1)         1(1)     350
505(505)      P2MP     1(1)         1(1)     0
506(506)      P2P      1(1)         1(1)     0
507(507)      P2P      1(1)         1(1)     0
508(508)      P2P      1(1)         1(1)     0
509(509)      P2P      1(1)         1(1)     0
510(510)      P2P      1(1)         1(1)     0
```

Enabling statistics on a bridge domain

Statistics may be enabled for a specific bridge domain.

NOTE

By default statistics are disabled on bridge domains. After enablement, statistics should be disabled when no longer needed because the collection of statistical information has a heavy impact on data traffic.

1. Enter the global configuration mode.

```
device# configure terminal
```

2. Enter the **bridge-domain** command to create a bridge domain at the global configuration level.

```
device(config)# bridge-domain 3
```

3. Enter the **statistics** command to enable statistics for all the logical interfaces and peers in the bridge domain.

```
device(config-bridge-domain-3)# statistics
```

NOTE
When statistics are no longer needed, use the **no statistics** command to disable statistics on the bridge domain.

The following example shows how to enable statistics on bridge domain 3.

```
device# configure terminal
device(config)# bridge-domain 3
device(config-bridge-domain-3)# statistics
```

The following example shows how to disable statistics on bridge domain 3.

```
device# configure terminal
device(config)# bridge-domain 3
device(config-bridge-domain-3)# no statistics
```

Displaying statistics for logical interfaces in bridge domains

Various commands can be used to display statistical information for bridge domains.

- Enter the **show statistics bridge-domain** command to display statistics for all logical interfaces and peers on all configured bridge domains.

```
device# show statistics bridge-domain

Bridge Domain 1 Statistics
Interface                    RxPkts          RxBytes          TxPkts          TxBytes
eth 0/1.100       821729          821729          95940360        95940360
eth 0/21.200      884484          885855          95969584        95484555
po  1.300           8884            8855              9684            9955

Bridge Domain 20 Statistics
Interface                    RxPkts          RxBytes          TxPkts          TxBytes
eth 0/6.400       821729          821729          95940360        95940360
eth 0/21.100        8884            8855              9684            9955
po  2.40          884484          885855          95969584        95484555
```

- Enter the **show statistics bridge-domain** command specifying a bridge-domain ID to view the statistics for a specific bridge domain. The following example displays statistics for bridge-domain ID 1.

```
device# show statistics bridge-domain 1

Bridge Domain 1 Statistics
Interface                    RxPkts          RxBytes          TxPkts          TxBytes
eth 0/1.100       821729          821729          95940360        95940360
eth 0/21.200      884484          885855          95969584        95484555
po  1.300           8884            8855              9684            9955
```

Clearing statistics on bridge domains

Statistical information can be cleared for all bridge domains or for a specific bridge domain.

- Enter the **clear statistics bridge-domain** command to clear statistics for all logical interfaces and peers on all configured bridge domains.

```
device# clear statistics bridge-domain
```

- Enter the **clear statistics bridge-domain** command specifying the bridge-domain ID to clear the statistics for a specific bridge domain. The following example shows how to clear statistics for bridge domain ID 1.

```
device# clear statistics bridge-domain 1
```

Multi-Chassis Trunking (MCT)

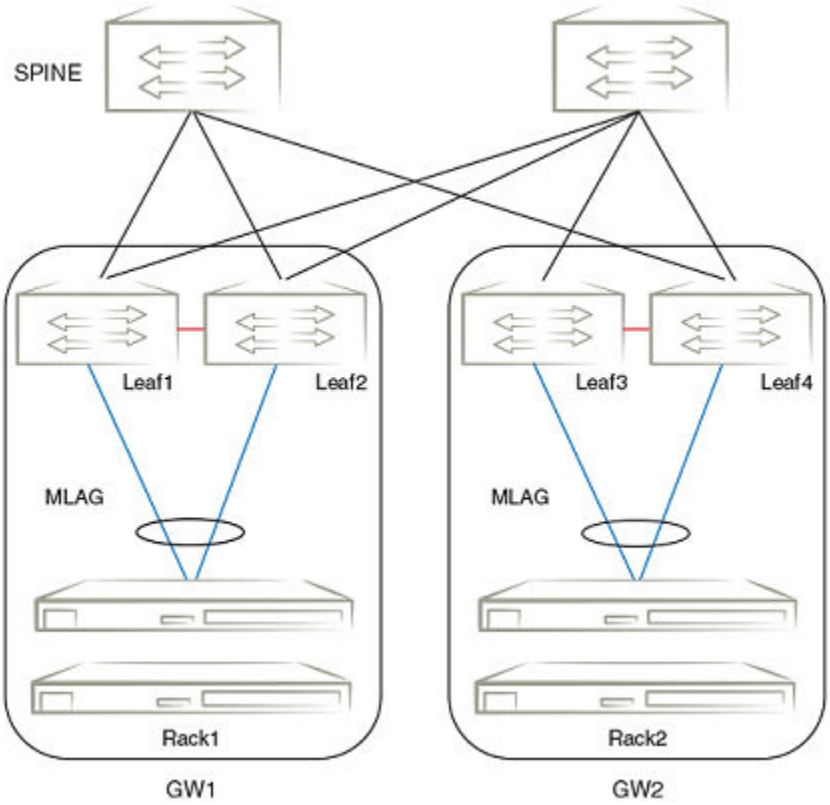
- MCT Overview....75
- Configuration considerations....82
- Configuring the BGP EVPN peer....83
- Configuring the MCT domain between a leaf switch pair....84
- Configuring additional MCT cluster parameters....88
- Configuring an auto-generated ESI for a cluster client....89
- Displaying MCT information....89
- Loop prevention in MCT through STP....91
- Bridge domain for Layer 2 multitenancy....94
- BFD support for Layer 3 protocols on MCT....97
- Enabling Layer3 routing for an MCT VLAN....98

MCT Overview

Multi-Chassis Trunking (MCT) is trunking that initiates at a single MCT-unaware server or switch and terminates at two MCT-aware switches. MCT allows the links to the two MCT-aware switches to appear to a downstream device as if they are coming from a single device on a single Link Aggregation (LAG) trunk interface or physical port.

In a datacenter network environment, LAG trunks provide link level redundancy and increased capacity. However, they do not provide switch-level redundancy. If the switch connected to the LAG trunk fails, the entire trunk loses network connectivity.

For SLX-OS MCT, the connected switches are MCT peer switches and communicate through an interface called a peer link. While the peer link's primary purpose is exchanging MCT control information between peer switches, it also carries data traffic from devices that are attached to only one MCT peer and have no alternative path. An MCT domain consists of the peer switches and the peer links that connect the switches.



SLX-OS Layer 2 MCT is based on RFC 7432 (BGP Ethernet VPN). The MP-BGP EVPN extension is the control plane on the SLX-OS device to perform both MAC synchronization and cluster management. MAC synchronization between the MCT peers synchronizes the MAC tables between the MCT nodes for node resiliency and faster convergence.

For the data plane, all data packets transmitted over a port connecting two MCT nodes are encapsulated with a Network Service Header (NSH).

Layer 3 protocols can be configured on a VE interface with an LAG as a switch port. Transmitted packet will use the local LAG port as an egress interface. If the local LAG port is down, packets are transmitted over the peer interface and will reach remote node using the remote LAG interface.

SLX-OS supports Multicast over MCT. For more information, see the *Extreme SLX-OS IP Multicast Configuration Guide*.

SLX-OS MCT uses the spanning tree protocol for loop prevention. For more information, see Loop prevention in MCT through STP on page 91.

MCT terminology

Before implementing MCT in your network, you must understand some key terms and definitions.

MCT peer devices	A pair of SLX-OS device configured as peers. The LAG interface is spread across two MCT peer devices and acts as the single logical endpoint to the MCT client.
MCT client	The MCT client is the device that connects with the MCT peer devices. It can be a switch or an endpoint server host in the single-level MCT topology or another pair of MCT switches in a multi-tier MCT topology.
MP-BGP EVPN extension	The control plane for Layer 2 MCT on the SLX-OS device.

MCT Cluster Client Edge Port (CCEP)	A port on one of the MCT peer devices that is a member of the LAG interface to the MCT client. To have a running MCT instance, at least one Link Aggregation Interface is needed with a member port on each peer device. While there is a LAG on the client device, CCEP on the MCT device can be a LAG or a physical port.
MCT Cluster Edge Port (CEP)	A port on MCT peer device that a member of a MCT VLAN and is not a Cluster Client Edge Port.
MCT VLANs	VLANs on which MCT clients are operating. These VLANs are explicitly configured in the MCT configuration.
MCT bridge domain	Bridge domain (BD) that is extended to the two MCT cluster nodes for multitenancy. Active-active forwarding occurs on traffic belonging to the BD.
Peer link	Interface through which MCT peer switches exchange MCT control information. It also carries data traffic from devices that are attached to only one MCT peer and have no alternative path.
MCT domain	Peer switches and the peer links that connect them.

MCT peer link

An MCT peer link is a physical interface or a channel group that connects two MCT peer switches.

> NOTE
> A peer link cannot be a Layer 3 interface.
> When the peer interface is configured, it is an internal switch port. An external switch port configuration on peer interface is not allowed.

As a tagged Layer 2 link, it carries packets for multiple VLANs. Only MCT member VLANs are carried over a peer link. For MCT member VLANs, MAC learning is disabled on the link. Each node learns the MAC addresses through Layer 2 forwarding which uses source MAC learning methodology and locally learned MACs are synced between the peers.

All data packets transmitted over the peer link are encapsulated with the Network service header (NSH) as the way to share metadata between devices. For forwarding traffic from the edge port over the peer link, the edge port VLANs must be configured as MCT member VLANs.

Cluster control VLAN

The cluster control VLAN is required to learn MAC addresses, resolve ARP for the BGP peer, and derive the outer MAC for the NSH tunnel. By default, the MCT cluster control vlan is 4090 and can be configured. If MCT is configured, other switch ports must not be part of the cluster control VLAN.

SLX-OS MCT control plane

Multiprotocol-BGP (MP-BGP) EVPN extension, as specified in RFC 7432, is used as the SLX-OS MCT control plane.

The control plane consist of the following components:

- EVPN instance—Mapped to a Layer 2 VLAN that the RFC refers to the VLAN-Based service interface.
- Ethernet segment ID (ESI)—10-byte integer that uniquely identifies the set of links connecting MCT leaf nodes to the client CE. By default, the ESI is auto generated.

  Optionally, the device supports for ESI auto generation based on Partners LACP system MAC address and LACP Port Key. This ESI is ESI type 1 as defined in RFC 7432 Section 5. The ESI is generated when the partner LACP MAC or port key is learned.
- MP-BGP route distinguisher (RD)—Encoded using RD type 1 as defined in RFC 4364 that consists of the following subfields:
  - 4-byte administrator subfield that is set with the 4-byte router ID
  - 2-byte assigned number subfield that is encoded with the all zeros for the Ethernet segment (ES) route, client ID for the Ethernet auto-discovery route, and EVPN ID (VLAN ID) for MAC and multicast routes.

- MP-BGP EVPN capability—When a BGP session is brought up to a MCT peer, BGP indicates to the peer that it is EVPN capable using BGP capability advertisement with the following information:
  - Capability code = 1 (MP-BGP)
  - AFI = 25 (L2VPN)
  - SAFI = 70 (EVPN)

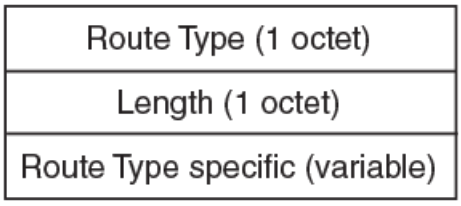
  If a BGP session already existed to the same peer, the existing BGP session is flapped to allow the advertisement of the EVPN capability.
- MP-BGP EVPN route types—Includes Ethernet Auto-Discovery (A-D), MAC/IP Advertisement, Inclusive Multicast Ethernet Tag, and Ethernet Segment routes.
- Network service header (NSH)—Inserted onto encapsulated packets or frames to provide a mechanism to carry shared metadata between network devices. The semantics of the shared metadata is communicated through the control plane to the participating nodes.
- Virtual Rbridge ID (VRB ID) assignment—In the NSH, the VRB ID is used for source suppression. Its 16-bit field is used to propagate the ESI label.
- Designated forwarder (DF)—A leaf node in a set of multi-homing leaf nodes connected to the same Ethernet segment that is elected for sending BUM traffic to a client for a VLAN ID on an Ethernet segment. By default, DF load balancing is disabled.

SLX-OS MCT operates in dual-homing mode.

MP-BGP EVPN Routes

RFC 7432 defines EVPN Network Layer Reachability Information (NLRI) with the format as shown in the following figure:

FIGURE 7 EVPN NLRI format



SLX-OS MCT supports the following route types.

TABLE 24 SLX-OS MCT route types

Route type	Route name	SLX-OS usage
1	Ethernet Auto-Discovery (per Ethernet Segment only)	Mass MAC withdraw and Designated forwarder election. The leaf node advertises one Ethernet A-D route for each client interface. When the client interface goes down, the leaf node withdraws the Ethernet A-D route which is served as a trigger for the remote leaf node to remove all MAC addresses learned over the affected client interface instead of withdrawing an individual MAC.
2	MAC/IP Advertisement	MAC synchronization of the MAC addresses between two MCT peers.
3	Inclusive Multicast Ethernet Tag	Advertisement of ingress replication usage and multicast label expected for each EVPN instance when receiving BUM traffic.
4	Ethernet Segment (ES)	Designated forwarder election. The ES route is used to update the remote peer when the MCT client is deployed and undeployed.

Network Service Header (NSH)

All data traffic passing through the MCT peer link is Network Service Header (NSH) encoded. The NSH is inserted onto encapsulated packets or frames to carry shared metadata between network devices.

The semantics of the shared metadata is communicated through the control plane to participating nodes, including classification information used for policy enforcement and a network context for forwarding post service delivery.

The NSH is transport independent and is carried in an overlay over existing underlays. After the NSH is added to a packet, an outer encapsulation is used to forward the original packet and the associated metadata. Transit network nodes forward the encapsulated packets as is.

NSH encapsulation examples

The following examples are for the Ethernet IPv4 packet and Layer 2 frame.

IPv4 Packet:

Outer Ethernet, ET=0x894F	NSH, NP= 0x1	original IP packet

L2 Frame:

Outer Ethernet, ET=0x894F	NSH, NP= 0x3	original frame

The following example is for the NHS in the SLX switch including the VRB ID used for source suppression.

NSH Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ver		O	C	R	R	R	R	R	R	Length						MD-type=0x1								Next Protocol							
Egress BD (16 Bit)																Service Index															
M	VRB Id (15 Bit)																														
Network Shared Context																															
Service Platform Context																															
Service Shared Context																															

The 16-bit VRB ID field is used to propagate the ESI label. Each cluster client represent one multi-chassis trunk link aggregation (MLAG) or ESI between the MCT peers. There is one-to-one mapping between the cluster client ID and the ESI. The cluster client ID is used to derive the VRB ID in NSH header for source suppression of BUM traffic. A maximum of 512 ESI or cluster clients are supported.

Designated forwarder election and load balancing

By default, DF load balancing is disabled. With load balancing disabled, one leaf node is the DF for all VLANs. The DF changes or a new DF is elected when the current DF leaf node goes down or its client interface is down. A non-DF leaf node that goes down does not impact the appointed DF. Also, any new node joining the Ethernet segment (ES) does not impact the appointed DF.

When DF load balancing is enabled, designated forwarder election can occur. To elect a designated forwarder (DF) for a VLAN ID on an ES, each leaf node exchanges its router IP address with its multi-homing leaf nodes through the ES route. The following algorithm uses the IP address to select the DF.

1. Upon the discovery of a new ES, a leaf node advertises the ES route and waits a default of three seconds for its peer to advertise the ES route.
2. When the timer expires, the leaf node builds a sorted list of leaf node IP addresses including its own address connected to the same ES.

3. The leaf node with the ordinal number that equals (V mod N) is elected the DF. V is the VLAN ID and N is the number of leaf nodes.
4. When the ES link fails, the leaf node withdraws its ES route which triggers the selection process to select a new DF. When a leaf node failure occurs, DF election is also triggered when the leaf node is up and down.

Designated forwarder election is triggered in the following scenarios:

- A client is deployed locally or remotely, or the BGP session comes up.

  The DF timer is started and the DF election is not performed until the timer expires.
- CCEP goes up or down.

  DF election is triggered as soon as ES/Enet-AD route from the remote peer is received or withdrawn to minimize the traffic loss.

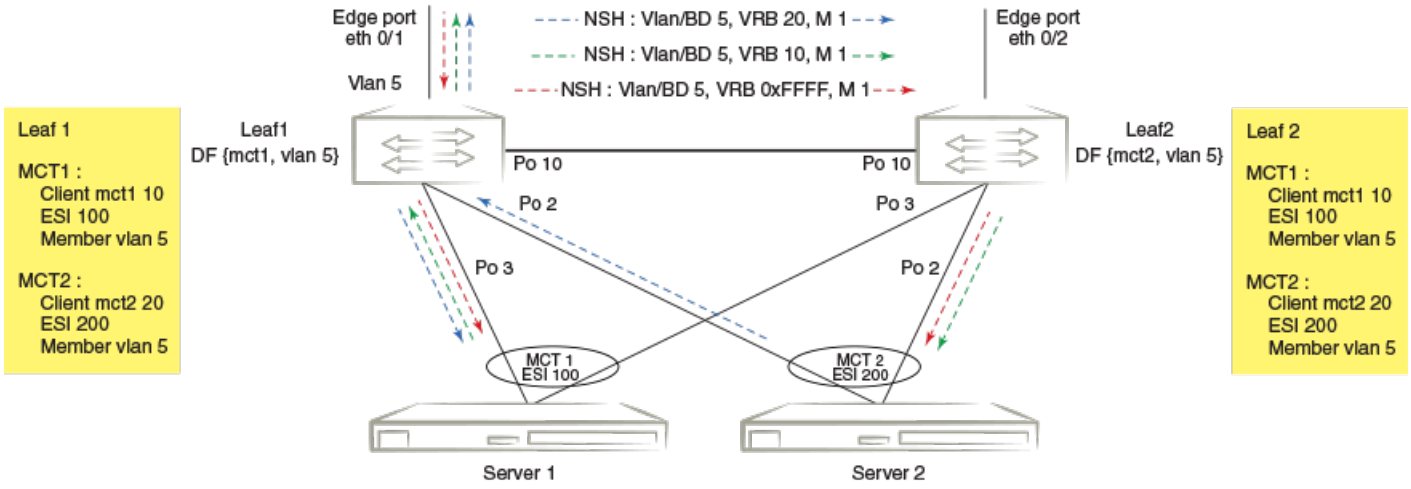
MCT data plane traffic forwarding

The MCT data plane provide the following traffic forwarding:

- BUM forwarding using NSH
- Unicast traffic forwarding

BUM forwarding using NSH

The following topology describes BUM forwarding using NSH.



Leaf1 and Leaf2 are MCT peers. MCT1 and MCT2 are two LAGs. On both leaf nodes, MCT1 is configured with client ID 10, ESI 100 and member VLAN 5. Similarly, MCT2 is configured with client ID 20, ESI 200, and member VLAN 5. In this example, Leaf1 is DF for {ESI 100, vlan 5} and Leaf2 is DF for {ESI 200, vlan 5}.

For the BUM traffic on the edge port, as indicated by the red arrows:

- In Leaf1, the BUM traffic received on edge port eth 0/1 is flooded based on the VLAN MGID. The packet transmitted over the peer link (Po10) is encapsulated with the NSH including the VLAN encoded with the same value as the incoming VLAN, the VRB ID set to the reserved value for the edge port (0xFFFE), and the M bit set to indicate the BUM packet.
- In the Leaf2, packet is flooded based on VLAN MGID.

For the BUM traffic on the client interface of the DF, as indicated by the green arrows:

- The BUM traffic received on client interface Po3 of MCT1, for which Leaf1 is DF, is flooded based on the VLAN MGID. The packet transmitted over peer link (Po10) is encapsulated with the NSH including the VLAN encoded with the same value as the incoming VLAN, the VRB ID set to 10 (the client ID of MCT1), and the M bit set to indicate the BUM packet.
- In Leaf2, the received VRB ID is used for source suppression and the packet is flooded based on VLAN MGID.

For the BUM traffic on the client interface of the non-DF node, as indicated by the blue arrows:

- The BUM traffic received on client interface Po2 of MCT2, for which Leaf1 is not the DF, is flooded based on the VLAN MGID. The packet transmitted over peer link (Po10) is encapsulated with the NSH including the VLAN encoded with same value as the incoming VLAN, the VRB ID set to 20 (the client ID of MCT2), and the M bit set to indicate the BUM packet.
- In Leaf2, the received VRB ID is used for source suppression and the packet is flooded based on the VLAN MGID.

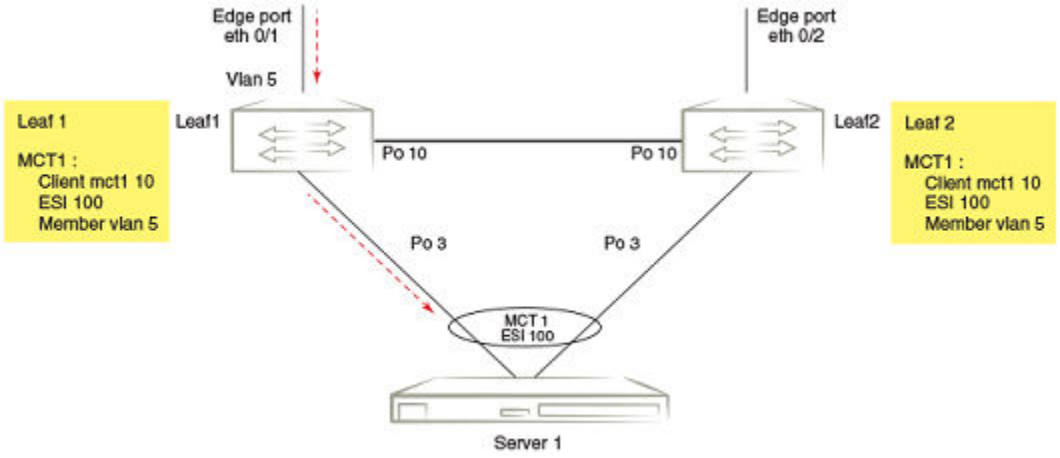
Unicast traffic forwarding

The following examples describe the following unicast forwarding traffic:

- Unicast local forwarding
- Unicast forwarding over the peer link from the edge port
- Unicast forwarding over peer link from client interface

Unicast local forwarding example

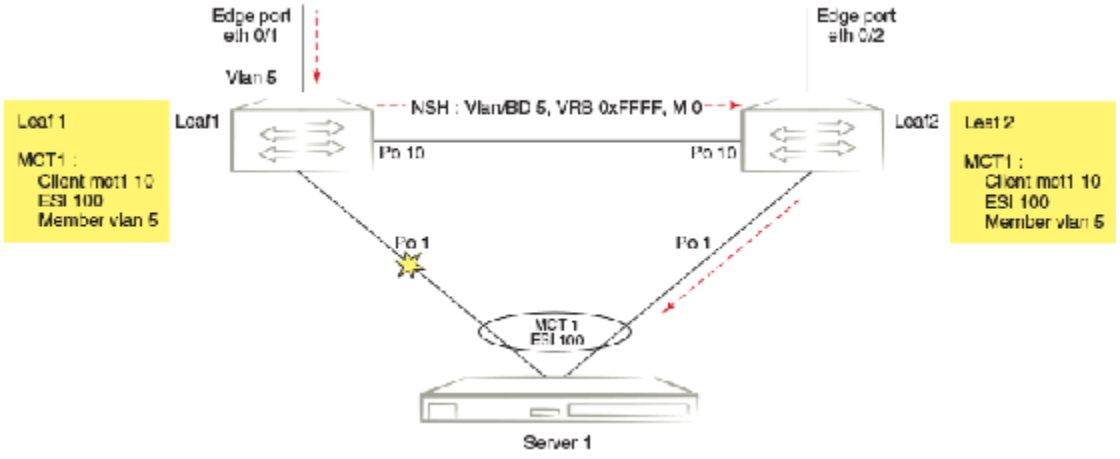
Unicast traffic sent between edge port eth 0/1 and Server 1 follows normal Layer 2 forwarding.



Unicast forwarding over the peer link from the edge port example

The unicast packet transmitted over the peer link (Po10) is encapsulated with the NSH including the VLAN encoded with the same value as the incoming VLAN, the VRB ID set to the reserved value for the edge port (0xFFFE), and the M bit set to zero to indicate the unicast packet.

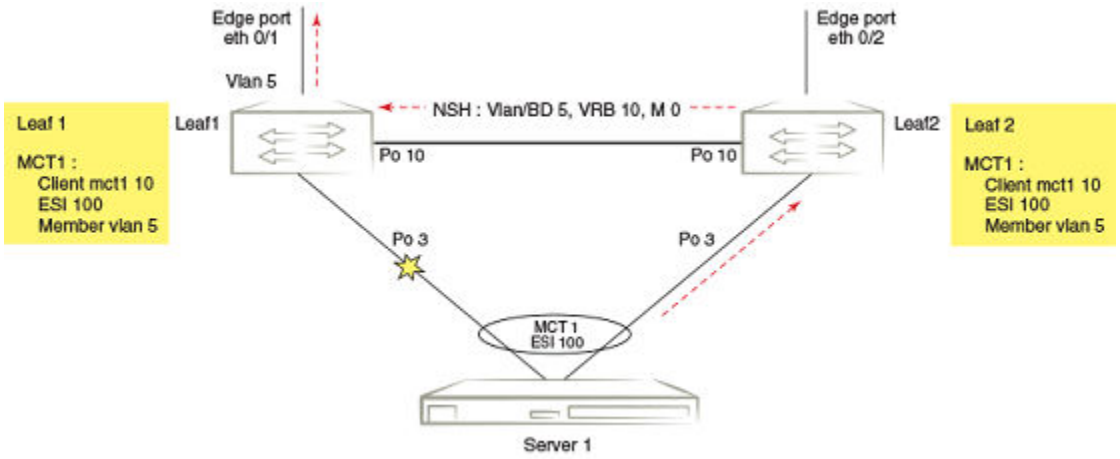
In Leaf2, the packet follows normal Layer 2 forwarding.



Unicast forwarding over peer link from client interface example

The unicast packet transmitted over the peer link (Po10) is encapsulated with the NSH including the VLAN encoded with same value as the incoming VLAN, the VRB ID set to 10 (the client ID of MCT1), and the M bit set to zero to indicate the unicast packet.

In Leaf1 packet follows normal Layer 2 forwarding.



Configuration considerations

- The SLX-OS device does not assume that the MCT peers are directly connected.
- A switch supports only one cluster.
- The cluster peer IP address is the IP address of the remote MCT node.

- The cluster peer interface is a directly connected port.
- Since the SLX-OS device supports both dynamic and static LAG between the MCT node and CE, it uses Ethernet segment identifier (ESI) type 0 regardless of the LAG type.
  - The ESI is auto generated.
  - You can configure ESI type 1 as defined in RFC 7432. The partner LACP system MAC address and LACP port key is used to automatically derive the ESI.
- You must configure and activate a BGP EVPN neighbor as the peer interface.
- The peer link must be configured and must be a switch port. When the peer interface is configured, it is an internal switch port. An external switch port configuration on peer interface is not allowed.
- The peer link will be part of all configured member VLANs.
- The cluster client ID for a specific ESI must be the same in both peers. The cluster client ID is used to derive VRB ID which is used in NSH header for source suppression.
- BGP Bidirectional Forwarding Detection (BFD) is used for the MCT peer keep alive.
- To form an MLAG, both MCT peers must use a unique LACP actor system ID and actor port key. The actor system ID is derived by appending a cluster ID to the MCT base system ID which is a device-defined value.

  System ID = mct_base_system_id (0180.c200) + cluster_id

  The actor port key is derived by appending a client ID to the MCT port key base which is a device-defined value.

  Key = MCT_LACP_KEY_BASE (3000) + client_ID
- The **no peer** command causes a controlled failover and the target node is removed permanently from the cluster. Primary and secondary controlled failover is supported, as is primary and secondary uncontrolled failover (on loss of heartbeat).

  If, for example, Node A is aware of its peer, Node B, and Node B is not aware of Node A, then Node A becomes the principal node. This can happen if the **no peer** command is executed on only a single node in the cluster.
- When the local configuration changes along with global configuration changes on either or both the nodes in a cluster, the nodes are not allowed to form the cluster. For example, when the nodes are disconnected, if the principal priority is changed on one or both the nodes and the global configuration, such as for overlay gateways, is configured on one or both the nodes, a configuration database mismatch occurs and the nodes are not allowed to form a cluster.

Configuring the BGP EVPN peer

Create a BGP EVPN address family to configure and activate the cluster EVPN peer. This configuration is associated with the MCT cluster peer configuration.

You also create an EVPN instance to add the bridge domains and VLANs associated with the MCT cluster.

1. In privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable BGP routing.

   ```
   device(config-terminal)# router bgp
   ```

3. Configure the EVPN peer with the autonomous system number (ASN).

   ```
   device(config-bgp-router)# neighbor 10.1.1.1 remote-as 100
   ```

4. Enter EVPN address family configuration mode

```
device(config-bgp-router)# address-family evpn
```

5. Activate the EVPN peer.

```
device(config-bgp-evpn)# neighbor 10.1.1.1 activate
```

6. Access global configuration mode.

```
device(config-bgp-router)# exit
```

7. Access EVPN configuration mode.

```
device(config)# evpn myinstance
```

8. Configure the bridge domains.

```
device(config-evpn-myinstance)# bridge-domain add 1-2000,4000-4096
```

   For more information on the bridge domain configuration, see Bridge domain for Layer 2 multitenancy on page 94.

9. Configure the VLANs.

```
device(config-evpn-myinstance)# vlan add 1-2000
```

10. Enable auto-generation of a route distinguisher (RD) for this EVPN instance.

```
device(config-evpn-myinstance)# rd auto
```

11. Enable the auto-generation of the import and export route-target community attributes and ignore the autonomous system (AS) number.

```
device(config-evpn-myinstance)# route-target both auto ignore-as
```

The following example shows the steps in the previous configuration.

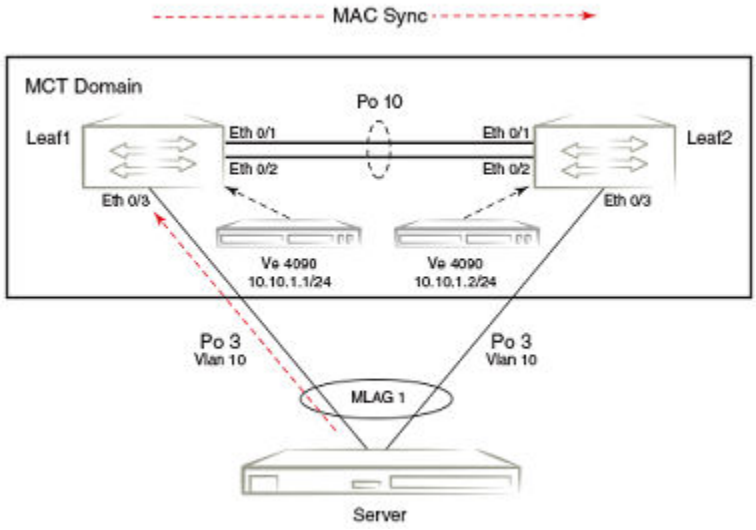
```
device# configure terminal
device(config-terminal)# router bgp
device(config-bgp-router)# neighbor 10.1.1.1 remote-as 100
device(config-bgp-router)# address-family evpn
device(config-bgp-evpn)# neighbor 10.1.1.1 activate
device(config-bgp-router)# exit
device(config)# evpn myinstance
device(config-evpn-myinstance)# bridge-domain add 1-2000,4000-4096
device(config-evpn-myinstance)# vlan add 1-2000
device(config-evpn-myinstance)# rd auto
device(config-evpn-myinstance)# route-target both auto ignore-as
```

Configuring the MCT domain between a leaf switch pair

Before configuring MCT domain between a leaf switch pair, ensure that the following configurations exist:

- Layer 2 interface for the cluster peer interface
- VLANs for the cluster members
- Port channel for Link Aggregation or an Ethernet interface as a client interface

Refer to the following figure for configuring the Leaf1 and Leaf2 clusters and clients.



Configuring the Leaf1 cluster and client

Perform the following steps to configure the Leaf1 cluster and client of the MCT switch pair.

1. In privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Create a cluster on the device.

```
device (config)# cluster MCT1 1
```

3. Configure the peer address.

```
device(config-cluster-1)# peer 10.1.1.2
```

   The peer address is the remote MCT node IP address. In this example, it is the address for leaf2.

   This address corresponds with the neighbor in BGP EVPN address family configuration for the peer.

4. Configure the peer interface.

```
device(config-cluster-1)# peer-interface port-channel 10
```

   The peer interface must be a valid Layer 2 interface. You should configure the peer interface before deploying the configuration.

   NOTE
   A peer link cannot be a Layer 3 interface.
   When the peer interface is configured, it is an internal switch port. An external switch port configuration on peer interface is not allowed.

5. Deploy the cluster.

```
device(config-cluster-1)# deploy
```

6. Create the client for the cluster and access cluster client configuration mode.

```
device(config-cluster-1)# client mlag1 1
```

   On both MCT nodes, you must configure the same client ID.

7. Configure the interface to the cluster client instance.

```
device(config-cluster-client-1)# client-interface port-channel 3
```

   The port channel specifies the LAG ID.

   The client interface can also be a physical interface, for example:

```
device(config-cluster-client-1)# client-interface Ethernet 0/5
```

   The client interface cannot be added under multiple client entries.

8. Optionally, configure the Ethernet Segment ID (ESI) type 1 as defined in RFC 7432.

```
device(config-cluster-client-1)# esi auto lacp
```

9. Deploy the cluster client.

```
device(config-cluster-client-1)# deploy
```

The following example is the steps in the previous configuration.

```
device# configure terminal
device (config)# cluster MCT1 1
device(config-cluster-1)# peer 10.1.1.2
device(config-cluster-1)# peer-interface port-channel 10
device(config-cluster-1)# deploy
device(config-cluster-1)# client mlag1 1
device(config-cluster-client-1)# client-interface port-channel 3
device(config-cluster-client-1)# esi auto lacp
device(config-cluster-client-1)# deploy
```

After configuring the Leaf1 cluster and client, configure the Leaf2 cluster and client.

Configuring the Leaf2 cluster and client

Perform the following steps to configure the Leaf2 cluster and client of the MCT switch pair.

1. In privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Create a cluster on the device.

```
device (config)# cluster MCT1 1
```

3. Configure the peer address.

```
device(config-cluster-1)# peer 10.1.1.1
```

   The peer address is the peer MCT node IP address. In this example, it is the address for Leaf1.

   This address corresponds with the neighbor in BGP EVPN address family configuration for the peer.

4. Configure the peer interface.

```
device(config-cluster-1)# peer-interface port-channel 10
```

   The peer interface must be a valid Layer 2 interface and is the same as the MCT peer. You should configure the peer interface before deploying the configuration.

   NOTE
   A peer link cannot be a Layer 3 interface.
   When the peer interface is configured, it is an internal switch port. An external switch port configuration on peer interface is not allowed.

5. Deploy the cluster.

```
device(config-cluster-1)# deploy
```

6. Create the client for the cluster and access cluster client configuration mode.

```
device(config-cluster-1)# client mlag1 1
```

   On both MCT nodes, you must configure the same client ID.

7. Configure the interface to the cluster client instance.

```
device(config-cluster-client-1)# client-interface port-channel 3
```

   The port channel specifies the LAG ID.

   The client interface can also be a physical interface, for example:

```
device(config-cluster-client-1)# client-interface Ethernet 0/5
```

   The client interface cannot be added under multiple client entries.

8. Optionally, configure the Ethernet Segment ID (ESI) type 1 as defined in RFC 7432.

```
device(config-cluster-client-1)# esi auto lacp
```

9. Deploy the cluster client.

```
device(config-cluster-client-1)# deploy
```

The following example is the steps in the previous configuration.

```
device# configure terminal
device (config)# cluster MCT1 1
device(config-cluster-1)# peer 10.1.1.1
device(config-cluster-1)# peer-interface port-channel 10
device(config-cluster-1)# deploy
device(config-cluster-1)# client mlag1 1
device(config-cluster-client-1)# client-interface port-channel 3
device(config-cluster-client-1)# esi auto lacp
device(config-cluster-client-1)# deploy
```

Configuring additional MCT cluster parameters

The SLX-OS device has additional cluster commands with default values. You can change the parameters for these commands in cluster configuration mode.

Changing the client-isolation mode

Isolation mode defines the action to be taken when the BGP control session goes down between the MCT nodes while the cluster is in deployed state. When the client-isolation mode is strict, the client interface will be shutdown.

By default, client-isolation mode is loose. In loose mode, both peers act as DF masters. When the EVPN control session goes down, the peer device performs the master/slave negotiation. After negotiation, the slave shuts down its peer ports, and the master peer ports continue to forward the traffic (keep-alive VLAN configured).

You can configure strict mode. In this mode, when the EVPN control session goes down, the interfaces on both the cluster devices are administratively shut down. In strict mode, the client is completely isolated from the network if the control session is not operational.

Use the **client-isolation-strict** command to configure the strict mode on both nodes, as shown in the following example.

```
device(config-cluster-1)# client-isolation-strict
```

Changing the designated-forwarder hold timer value

Upon expiration of the designated-forwarder hold timer, the reelection of the designated forwarder is considered.

By default, the hold time is three seconds. Use the **designated-forwarder-hold-time** command to change the time in seconds from 1 to 60 seconds, as shown in the following example.

```
device(config-cluster-1)# designated-forwarder-hold-time 35
```

Enabling DF load balancing

By default, DF load balancing is disabled. When DF load balancing is disabled, DF election is triggered only when the current DF leaf node goes down or its client interface is down. When a non-DF leaf node goes down or a new node joins the ES, DF election is not triggered.

When DF load balancing is enabled, the DF election is triggered in the following scenarios:

- A client is deployed locally or remotely.
- The BGP cluster control protocol (CCP) session comes up.
- Remote CCEP goes up or down.

Use the **df-load-balance** command to enable DF load balancing, as shown in the following example.

```
device(config-cluster-1)# df-load-balance
```

Changing the cluster control VLAN

The cluster control VLAN is required for MAC learning, resolving ARP for the BGP peer, and to derive the outer MAC address for the NSH tunnel.

By default, the control VLAN is 4090. You can configure a value from 1 to 4090. Use the **cluster-control-vlan** command, as shown in the following example.

```
device(config-cluster-1)# cluster-control-vlan 20
```

NOTE
If MCT is configured, other switch ports must not be part of the cluster control VLAN.

Moving the traffic from an MCT node to the remote node

Use the **client-interfaces-shutdown** command to move all the traffic on the node to the remote MCT node by disabling the local client interfaces administratively, as shown in the following example.

```
device(config-cluster-1)# client-interfaces-shutdown
```

Configuring an auto-generated ESI for a cluster client

When a client interface is a port channel and LACP is running on the port channel, you can configure an automatically generated ESI value, as defined in RFC 7432 on the cluster client.

The following example shows how to configure an auto-generated ESI for the cluster client.

```
device(config)# cluster MCT1 1
device(config-cluster-1)# client MCT1-client 200
device(config-cluster-client-200)# esi auto lacp
```

Displaying MCT information

You can display detailed MCT information and related MCT MAC addresses.

To display the EVPN neighbor information, use **show ip bgp neighbors** command. This information includes the peer configured for the EVPN address family, the undeplyed MCT cluster, and the negotiation of the EVPN address family.

Displaying the cluster information

The following example shows the information of the cluster on the SLX-OS device.

```
device# show cluster 1

Cluster MCT1 1
==============
Cluster State: Deploy
Client Isolation Mode: Loose
DF Hold Time: 3
Configured Member Vlan Range: 2, 4-7
Active Member Vlan Range: 2, 4-7
Cluster Control Vlan: 4090
Configured Member BD Range:
Active Member BD Range:
No. of Peers: 0
No. of Clients: 0

 Peer Info:
 ----------
 Peer IP: 10.10.10.20, State: Up
```

```
Peer Interface: Ethernet 0/1
ICL Tunnel Type: NSH, State: Up

Client Info:
------------
Name     Id   ESI                           Interface    Local/Remote State
access1  100  0:11:22:33:80:0:0:0:0:0        Eth 0/3         Up/UP
access2  200  0:11:22:33:81:0:0:0:0:0        po-chan-2       Dep/UnDep
access3  300  0:11:22:33:82:0:0:0:0:0        Eth 0/8         Up/Down
```

Displaying the cluster client information

The following example displays client 100 information for cluster 10.

```
device# show cluster 10 client 100

Client Info:
============
Client                : access1
Client-id             : 100
Client state          : Undeployed
Interface state       : Up
Interface             : Ethernet 0/8
Vlans                 : 1-10, 100
Bridge Domains        :
Number of DF Vlans    : 0
Elected DF for vlans  :
Number of DF Bridge Domains   : 0
Elected DF for Bridge Domains :
```

Displaying and clearing the MAC address table cluster information

The following example displays the MCT cluster information in the MAC address table.

```
device# show mac-address-table cluster 1
Type Code - CL:Cluster Local MAC   CCL:Cluster Client Local MAC
            CR:Cluster Remote MAC  CCR:Cluster Client Remote MAC
VlanId/BDId   Mac-address       Type        State      Ports/LIF/PW
10 (V)        0000.1111.1111    CCR         Active     Eth 0/43
10 (V)        0000.1111.1112    Dynamic-CCL Active     Eth 0/43
20 (V)        0000.1111.1113    Dynamic-CL  Active     Eth 0/44
20 (V)        0000.1111.1114    CR          Active     Eth 0/10
4090 (V)      609c.9f5b.1301    Dynamic     Active     Eth 0/10
Total MAC addresses    :  5
```

You can also view the MAC entries for a specific client.

Clearing the MCT cluster MAC table entries

You can clear all cluster entries from the MAC address table or the entries for a specified client. The following example clears the MAC entries for client 3 of cluster 1.

```
device# clear mac-address-table cluster 1 client 3
```

Only the local MAC entries are deleted from the current node. Individual MAC withdrawal flush messages are sent through the EVPN. However, BGP still batches multiple routes to the remote node.

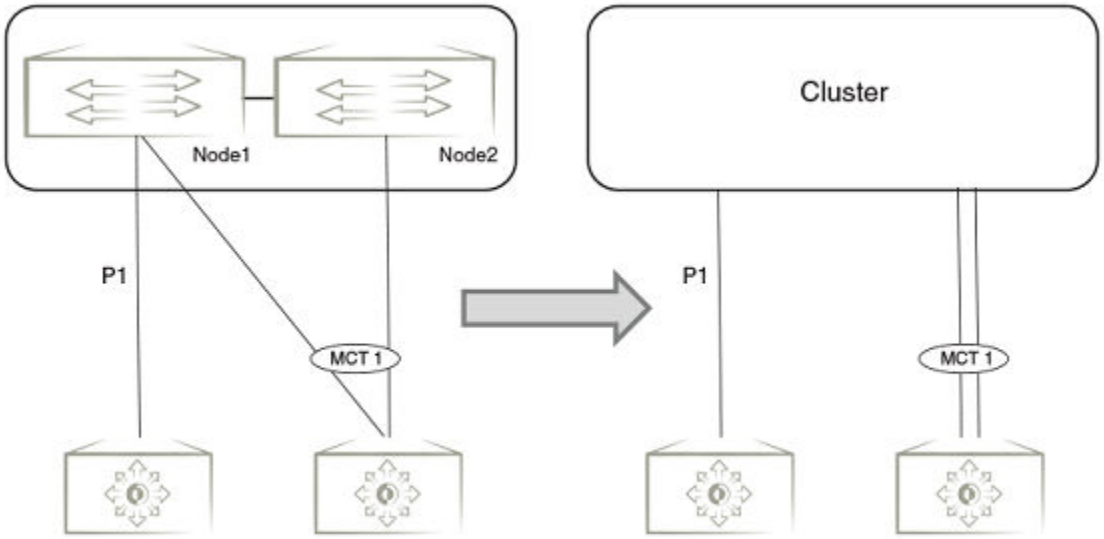
When the remote MCT peer receives the MAC withdrawal message, it only deletes the remote MAC entry. To clear MAC addresses on both nodes, you must issue **clear mac-address-table** commands on both MCT nodes.

Loop prevention in MCT through STP

In a cluster deployment, a loop situation may occur due to misconfigurations or any faulty connection in the network. When you enable spanning tree protocol (STP) on the SLX-OS device, the device uses STP as the mechanism for preventing loops in MCT interfaces.

NOTE
For more information on STP, refer to the "802.1d Spanning Tree Protocol" chapter.

The SLX-OS device treats the cluster as one virtual xSTP bridge with xSTP running between the cluster or standalone nodes.



- Each cluster has a unique bridge ID and priority.
- STP is enabled on the cluster edge ports connecting to other cluster or standalone nodes.

NOTE
STP is supported on the cluster edge ports only. STP cannot be enabled on peer link ports connecting the nodes in the cluster. STP does not update the port state of peer link ports.

Each node runs the spanning tree instance in a distributed manner. This instance considers all edge ports and the best information from the remote node to arrive at the spanning tree topology.

Each node updates the other member about its best information for each spanning tree instance and maintains a table of this information. This table is identical across all nodes in the cluster. The table information is used to determine the roles and states for the local edge ports. Thus, each node considers the port roles and states of the other node to arrive at a final spanning tree topology.

Bridge ID

In SLX-OS, when the cluster mode is disabled, the physical MAC address is used as the bridge identifier (ID). When cluster mode is enabled, the virtual MAC address is used as the bridge ID .

The bridge ID or switch MAC used in all nodes participating in the STP domain must have a unique MAC address. A cluster should act as a single node for all peer bridges in the STP topology. Since any of the switches in the cluster can be swapped out and moved to another cluster, the bridge ID does not belong to any piece of hardware.

Virtual bridge ID format

The bridge ID is derived from the reserved Extreme MAC address (Extreme OUI). The Extreme OUI is 00:E0:52.

The bridge ID has the following form:

<Extreme OUI (3bytes)>: <Cluster ID (2bytes)>:00

For example, the bridge identifier for a cluster with an ID as 2 will have the following ID:

00:E0:52:00:02:00

By default, the last byte is 0. However, it is configurable.

Bridge ID collision

The possibility of a virtual bridge ID collision can occur if bridges in the spanning-tree topology have the same cluster ID. If this exists, you can avoid collisions by using the **cluster-system-id** command to change the last byte of the virtual bridge ID. For more information, refer to the "Avoiding bridge ID collisions".

Port ID

In spanning tree, each port participating in the tree is assigned an ID which is the direct port number. In a cluster environment, this port number must be unique across both nodes. Also, for MCT interfaces, the port ID must be the same in both cluster nodes.

The port ID used in BPDUs is 16 bits long with two parts:

- 4 bits for port priority
- 12 bits for the port number. Port number allocation for MCT interfaces, regular LAG interfaces and Ethernet interfaces is done in these 12 bits.
  - Bit 11—MCT Type bit. For MCT interfaces, this bit is set. The MCT client ID will be encoded in rest of the 11 bits (Bit 10:0).
  - Bit 10—Node ID. When the cluster is enabled, this bit is set. The node ID is either 0 or 1. The cluster node with highest peer IP address is 1 and the other node is 0.
  - Bit 9—LAG Type bit. For LAG interfaces with non MCT interfaces, this bit is set. The LAG ID is encoded in the rest of the 9 bits (Bit 8:0).
  - Bit 8:0—Port number. The port number is encoded in these 9 bits.

This static encoding of the port IDs ensures that the ID is unique across the cluster and no two ports will have the same port ID.

Loop prevention configuration considerations.

- By default global spanning tree is disabled in the SLX-OS device. When the global spanning tree configuration is enabled, all switchports are enabled with spanning tree. By default, the physical MAC is used as the bridge ID and the port ID computation is local mode. However, when the cluster is deployed, the virtual MAC address is used as the bridge ID.
- Mismatched global configuration of xSTP across different nodes in cluster is not supported. For example, if one of the nodes in a cluster is configured for RSTP and the another one is configured for STP or MSTP, this is not supported.
- Spanning tree protocol can be enabled on the switch by the **protocol spanning-tree** command. An interface starts to participate in the spanning tree once it is configured by the **no spanning-tree shutdown** command.
- STP in MCT is supported on the cluster edge ports only. You cannot enable STP on a peer link.
- If STP is already enabled when changing the cluster, there will be a change in the bridge ID and port ID that can cause a possible re-convergence. Configure the cluster before enabling STP to avoid any re-convergence.

Configuring the last byte of the bridge ID

The bridge ID (Switch MAC) used by all nodes participating in the STP domain should have a unique MAC address. You can configure the last byte of the bridge ID to ensure that two or more clusters do not have the same bridge IDs in a Layer 2 topology.

Before changing the last byte of the bridge ID, the cluster must be configured.

Ensure that the global STP configuration is consistent on both cluster nodes. Configuration mismatch handling is not supported.

1. In privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable a version of STP globally and access its configuration mode.

   ```
   device(config)# protocol spanning-tree rstp
   ```

   In this step, RSTP is enabled.

3. Change the last byte on the bridge ID to a unique value.

   ```
   device(config-rstp)# cluster-system-id 2
   ```

   By default, the value of the last byte is 0. Enter an integer from 1 to 255.

4. Verify the change to the bridge ID.

   ```
   device(config-rstp)# do show spanning-tree brief
   ...
   Bridge ID Priority 32768
             Address 00e0.5200.0102
             Hello Time 2, Max Age 20,Forward Delay 15, Tx-HoldCount6
             Migrate Time 3 sec
   ...
   ```

5. Repeat the previous steps on the other node in the cluster. You must configure the cluster nodes with the same value.

The following example show the previously configured steps.

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(config-rstp)# cluster-system-id 2
device(config-rstp)# do show spanning-tree brief
```

Displaying the root port on the cluster

In the following example, the **show spanning-tree brief** command displays the root port (RTPT), Ethernet 0/22, is on the current node in the cluster.

```
device# show spanning-tree brief
     Spanning-tree Mode: Rapid Spanning Tree Protocol
     Root ID    Priority 32768
                Address 0005.1ecd.0b8a
                Hello Time 2, Max Age 20, Forward Delay 15
                Root Port ID: Eth 0/22
     Bridge ID  Priority 32768
                Address 0105.3352.6f28
                Hello Time 2, Max Age 20, Forward Delay 15, Tx-HoldCount 6
                Migrate Time 3 sec
Interface  Role   Sts   Cost   Prio   Link-type   Edge
-----------------------------------------------------------------
Eth 0/1    DES    FWD   2000   128    P2P         No
Eth 0/2    DES    FWD   2000   128    P2P         No
Eth 0/22   RTPT   FWD   2000   128    P2P         No
```

If the root port is on another node in the cluster, the **show spanning-tree brief** command displays the following output for Ethernet 0/23 as the alternate (ALT) port.

```
# show spanning-tree brief
     Spanning-tree Mode: Rapid Spanning Tree Protocol
     Root ID    Priority 32768
                Address 0005.1ecd.0b8a
                Hello Time 2, Max Age 20, Forward Delay 15
                Root Port ID : Eth 0/22 (Remote)
     Bridge ID  Priority 32768
                Address 0105.3352.6f28
                Hello Time 2, Max Age 20, Forward Delay 15, Tx-HoldCount 6
                Migrate Time 3 sec
Interface  Role   Sts   Cost   Prio   Link-type   Edge
-----------------------------------------------------------------
Eth 0/1    DES    FWD   2000   128    P2P         No
Eth 0/2    DES    FWD   2000   128    P2P         No
Eth 0/23   ALT    DSC   2000   128    P2P         No
```

Bridge domain for Layer 2 multitenancy

A bridge domain (BD) is a generic broadcast domain that contains logical interfaces of Layer 2 interfaces and MCT interfaces with different VLAN classifications.

> NOTE
> A BD broadcast domain is different from a VLAN broadcast domain.

You must create a logical interface with specific VLANs on an Ethernet or port channel interface. Then, the logical interface is added to the bridge domain to bind them.

When the BD is created, a unique internal VLAN ID (IVID) is allocated for the BD by the device. The IVID allows only the logical interfaces belonging to the BD to forward traffic with each other. The flooding and forwarding of the Layer 2 traffic is restricted to the broadcast domain of the BD.

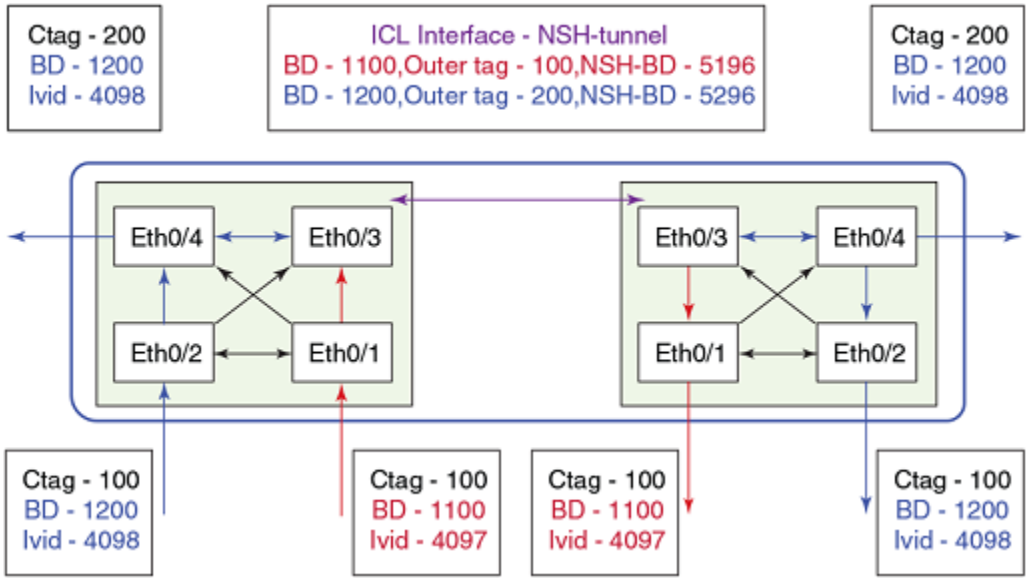
The SLX-OS device supports multitenancy for a bridge domain in an MCT cluster deployment. The BD traffic is extended to a node through the ICL using NSH tunnel encapsulation. The BD is mapped to an NSH BD sent across the ICL.

On the other node, traffic on this NSH BD, is classified to the BD IVID. Based on the configuration for this BD on this node, the traffic may be forwarded or flooded in this IVID.

MCT bridge domain example

In the following example, a dot1q VLAN tag (ctag) 100 is required by two customers. Two BDs are required.

- BD 1100—The device gives a unique IVID 4097 and maps ctag 100 and 200 traffic to this BD.
- BD 1200—The device gives a unique IVID 4098 and maps ctag 100 and 200 traffic to this BD.



For customer 1, nodes, 1 and 2 are configured with ctag 100 on the respective eth0/1 interface. A logical interface on eth0/1 is configured for VLAN 100 and is added to BD 1100.

For customer 2, nodes 1 and 2 are configured with the same ctag 100 on eth0/2 and ctag 200 on eth0/4, and added to BD 1200. A logical interface on eth0/2 is configured for VLAN 100 and is added to BD 1200. A logical interface on eth0/4 is configured for VLAN 200 and is added to BD 1200.

When the BD is added as a member to an MCT cluster, the traffic for each BD is carried to the other node using NSH BD in the NSH encapsulation. On the receiving node, the NSH BD is mapped to the configured BD.

Configuration considerations for bridge domain for MCT

- The BD must contain the CCEP interfaces and the peer interface.
- The peer interface can be a physical port or a port-channel.
- The peer interface carries only the MCT BD traffic.
- The logical interface is not created for the peer interface.
- Only the unknown flood traffic hits the flood multicast group ID(MGID) of MCT BD and is flooded. The learnt traffic hits the forwarding database (FDB). The FDB points to the CCEP logical interface (LIF) and the traffic goes to the egress port of the CCEP LIF.
- The egress port of the CCEP LIF is either the CCEP interface or the peer interface based on the availability of the link.

Configuring a bridge domain

A bridge domain is a broadcast domain for the forwarding and flooding of Layer 2 traffic restricted to the domain of the BD. After you configure the BD for Layer 2 multitenancy, you can add it as a member to an MCT cluster.

Prior to completing the following task, the Ethernet logical interface and VLANs must be created. There is an example at the end of this task that shows all the steps in order.

You can configure a bridge domain by completing the following task.

1. From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Create a bridge domain.

```
device(config)# bridge-domain 50
```

   In this example, bridge domain 50 is created as a multipoint service. By default, the bridge-domain service type is multipoint.

3. Bind the logical interfaces for attachment circuit endpoints to the bridge domain.

```
device(config-bridge-domain-50)# logical-interface port-channel 52.5
```

   In this example, the port-channel logical interface 52.5 is bound to bridge domain 50. You can also bind an Ethernet interface.

4. Repeat the previous step to bind additional logical interface to the bridge domain.

```
device(config-bridge-domain-50)# logical-interface port-channel 110.55
```

5. Enter Privileged EXEC mode.

```
device(config-bridge-domain-50)# end
```

6. (Optional) Display the running configuration of the bridge domain.

```
device# show running-config bridge-domain
bridge-domain 50 p2mp
logical-interface port-channel 110.55
logical-interface port-channel 52.5
!
```

7. (Optional) Display the information about the configured bridge domain.

```
device# show bridge-domain
Bridge-domain 50
------------------------------
Bridge-domain Type: MP
Number of configured end-points: 2 , Number of Active end-points: 2
VE if-indx: 0, bpdu-drop-enable: TRUE
mac-limit: 0
VLAN: 5, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: po52.5
Un-tagged Ports:
VLAN: 4093, Tagged ports: 0(0 up), Un-tagged ports: 1 (1 up)
Tagged Ports:
Un-tagged Ports: po110.55
```

The following example shows the creation of a logical interface and VLAN in addition to the bridge domain configuration.

```
device# configure terminal
device(config)# interface port-channel 52
device(config-Port-channel-52)# switchport
device(config-Port-channel-52)# switchport mode trunk
device(config-Port-channel-52)# logical-interface port-channel 52.5
device(conf-if-po-lif-52.5)# vlan 5
device(conf-if-po-lif-52.5)# exit
device(config-Port-channel-52)# exit
device(config)# interface port-channel 110
device(config-Port-channel-110)# switchport
device(config-Port-channel-110)# switchport mode trunk-no-default-native
device(config-Port-channel-110)# logical-interface port-channel 110.55
device(conf-if-po-lif-110.55)# untagged
device(conf-if-po-lif-110.55)# exit
device(config-Port-channel-110)# exit
device(config)# bridge-domain 50
device(config-bridge-domain-50)# logical-interface port-channel 52.5
device(config-bridge-domain-50)# logical-interface port-channel 110.55
```

Enabling and displaying bridge domain statistics

By default, bridge domain statistics are disabled on the device. Follow this procedure to enable statistics on a bridge domain and display its statistics.

1. Enter the global configuration mode.

```
device# configure terminal
```

2. Access the configuration mode for the bridge domain.

```
device(config)# bridge-domain 50
```

3. Enable statistics for the bridge domains.

```
device(config-bridge-domain-50)# statistics
```

4. Display the statistics for the bridge domain.

```
device(config-bridge-domain-50)# do show statistics bridge-domain
Bridge-Domain Statistics
BD Index      Rx Pkts              Rx Bytes             Tx Pkts              Tx Bytes
50            0                    0                    0                    0
```

Clearing the bridge domain statistics

Use the **clear statistics bridge-domain** command to clear the statistics for the bridge domains.

```
device# clear statistics bridge-domain
```

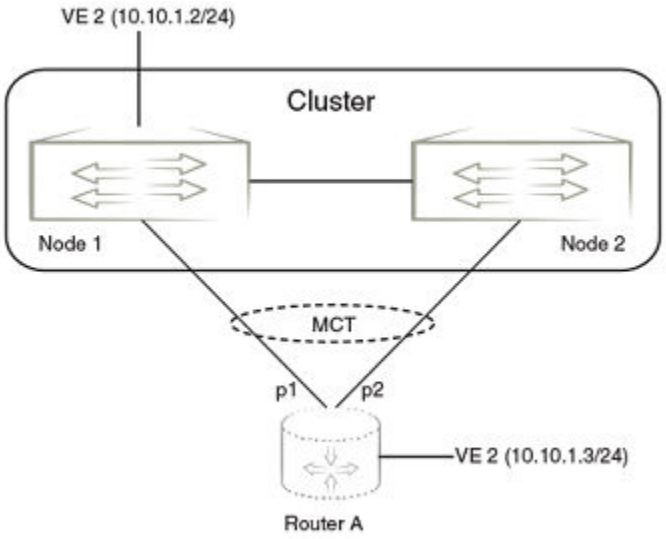
To clear the statistics for a specific bridge domain, specify the bridge domain with the **clear statistics bridge-domain** command.

```
device# clear statistics bridge-domain 1
```

BFD support for Layer 3 protocols on MCT

Layer 3 BFD sessions are supported on MCT interfaces used by SVI sessions.

For more information on BFD, refer to the *SLX-OS Layer 3 Configuration Guide*. The following figure shows BFD on MCT.



BFD packet transmission

Cluster Node 1 hosting the BFD session starts the BFD packet transmission over the local member port p1. If the local member port goes down, the parent VE interface of MCT is in the operationally UP state and is reachable using the MCTclient interface of cluster Node 2. In case all local MCT client ports go down, the BFD packets is the transmitter through the peer interface to the remote cluster node. Also, the remote node forwards the BFD packet through its local MCT client interface.

BFD packet reception

The BFD packet received in the MCT member ports of the remote Node 1 is redirected to the destination Node 1 through the peer interface.

Enabling Layer3 routing for an MCT VLAN

Layer 3 routing is supported for IPv4 and IPv6 BGP, and OSPF routing protocols on an MCT VLAN.

The enabling of the Layer 3 protocols on the MCT VLAN are the same as enabling them on a VE interface. You must first create the VE interface for an MCT VLAN.

The following configuration example enables OSPFv2 and OSPFv3 protocols on VE 200 for the MCT member VLAN 2.

```
router ospf
  area 0

ipv6 router ospf
  area 0

vlan 2
 router-interface Ve 200

interface Ve 200
 ipv6 address 2001::1/64
 ip address 10.2.2.1/24

 ip ospf area 0
 ipv6 ospf area 0
```

```
 !
 no shutdown
!
```

802.1d Spanning Tree Protocol

- Spanning Tree Protocol overview 101
- Spanning Tree Protocol configuration notes 101
- STP features 104
- STP parameters 106
- Configuring STP 108

Spanning Tree Protocol overview

The Spanning Tree Protocol (STP) prevents Layer 2 loops in a network by providing redundant links. If a primary link fails, the backup link is activated and network traffic is not affected. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs.

The IEEE 802.1d Spanning Tree Protocol (STP) runs on bridges and switches that are 802.1d-compliant.

These variants are Rapid STP (RSTP), Multiple STP (MSTP), Per-VLAN Spanning Tree Plus (PVST+), and Rapid-PVST+ (R-PVST+)

When the spanning tree algorithm is run, the network switches transform the real network topology into a spanning tree topology. In an STP topology any LAN in the network can be reached from any other LAN through a unique path. The network switches recalculate a new spanning tree topology whenever there is a change to the network topology.

For each LAN, the switches that attach to the LAN select a designated switch that is the closest to the root switch. The designated switch forwards all traffic to and from the LAN. The port on the designated switch that connects to the LAN is called the designated port. The switches decide which of their ports is part of the spanning tree. A port is included in the spanning tree if it is a root port or a designated port.

STP runs one spanning tree instance (unaware of VLANs) and relies on long duration forward-delay timers for port state transition between disabled, blocking, listening, learning and forwarding states.

Spanning Tree Protocol configuration notes

Enabling the Spanning Tree Protocol (STP) creates a loop-free topology of Ethernet LANs connected by bridge devices.

The Extreme device supports STP as described in the IEEE 802.1d-1998 specification.

The STP is disabled by default on the Extreme device. Thus, any new VLANs you configure on the Extreme device have STP disabled by default.

Optional features

The following STP configuration features are optional:

- Root guard
- BPDU guard
- PortFast

STP states

Each Layer 2 interface participating in a spanning tree is in one of five states.

A network topology of bridges typically contains redundant connections to provide alternate paths in case of link failures. The redundant connections create a potential for loops in the system. As there is no concept of time to live (TTL) in Ethernet frames, a situation may arise where there is a permanent circulation of frames when the network contains loops. To prevent this, a spanning tree connecting all the bridges is formed in real time.

Every Layer 2 interface running the STP is in one of these states:

State	Action or inaction
Blocking	The interface does not forward frames. Redundant ports are put in a blocking state and enabled when required. This is a transitional state after initialization.
Listening	The interface is identified by the spanning tree as one that should participate in frame forwarding. This is a transitional state after the blocking state for a legacy STP.
Learning	The interface prepares to participate in frame forwarding. This is a transitional state after the blocking state for a legacy STP.
Forwarding	The interface forwards frames. This is a transitional state after the learning state.
Disabled	The interface is not participating in a spanning tree because of shutdown of a port or the port is not operationally up. Any of the other states may transition into this state.

BPDUs

To build a spanning tree for the bridge topology, the bridges must exchange control frames called Bridge Protocol data units (BPDUs).

To construct a spanning tree requires knowledge of the all the participants. The bridges must determine the root bridge and compute the port roles (root, designated, or blocked) with only the information that they have. To ensure that each bridge has enough information, the bridges use BPDUs to exchange information about bridge IDs and root path costs.

A bridge sends a BPDU frame using the unique MAC address of the port itself as a source address, and a destination address of the STP multicast address 01:80:C2:00:00:00.

BPDUs are exchanged regularly (every 2 seconds by default) and enable switches to keep track of network changes and to start and stop forwarding through ports as required.

When a device is first attached to a switch port, it does not immediately forward data. It instead goes through a number of states while it processes inbound BPDUs and determines the topology of the network. When a host is attached, after a listening and learning delay of about 30 seconds, the port always goes into the forwarding state. The time spent in the listening and learning states is determined by the forward delay. However, if instead another switch is connected, the port may remain in blocking mode if it would cause a loop in the network.

There are four types of BPDUs in the original STP specification:

- Configuration BPDU (CBPDU) is used for spanning tree computation.
- Topology Change Notification (TCN) BPDU is used to announce changes in the network topology.
- RSTP BPDU is used for RSTP
- MSTP BPDU is used for MSTP

TCN BPDUs

TCN BPDUs are used to inform other switches of port changes.

TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

Consider these configuration rules:

- TCN BPDUs are sent per VLAN.
- TCN BPDUs are sent only in those VLANs in which a topology change is detected.
- TCN BPDUs are sent only in those VLANs for which the bridge is not the root bridge.
- If a topology change is detected on a VLAN for which the bridge is the root bridge, the topology change flag is set in the configuration BPDU that is sent out.

For a given link, in conjunction with the configuration rules, a TCN BPDU is sent out as follows:

- On an access port, only a standard IEEE TCN BPDU is sent out. This TCN BPDU corresponds to a topology change in the access VLAN.
- On a trunk port, if VLAN 1 is allowed (either untagged or untagged), a standard IEEE TCN BPDU is sent for VLAN 1.
- On a trunk port, if the native VLAN is not 1, an untagged TCN BPDU is sent to Cisco or Extreme proprietary MAC address for that VLAN.
- On a trunk port, a tagged TCN BPDU is sent to Cisco or Extreme proprietary MAC address for a tagged VLAN.

As part of the response to TCN BPDUs, the Topology Change and Topology Change Acknowledgment flags are set in all configuration BPDUs corresponding to the VLAN for which the TCN was received.

When a topology change is detected on a trunk port, it is similar to detecting topology changes in each VLAN that is allowed on that trunk port. TCN BPDUs are sent for each VLAN as per the rules.

STP configuration guidelines and restrictions

Follow these configuration guidelines and restrictions when configuring STP and STP variants:

- Only one form of a standing tree protocol, such as STP or RSTP, can be enabled at a time. You must disable one form of xSTP before enabling another.
- When any form of STP is enabled globally, that form of STP is enabled by default on all switch ports.
- LAGs are treated as normal links for any form of STP.
- The STP is disabled by default on the SLX device. Thus, any new VLANs you configure on the SLX device have STP disabled by default.
- PVST/RPVST BPDUs are flooded only if PVST/RPVST is not enabled. STP/RSTP (IEEE) BPDUs are never flooded if STP/RSTP is not enabled.

Understanding the default STP configuration

You should be familiar with STP defaults before you make configuration changes.

TABLE 25 Default STP configuration

Parameter	Default setting
Spanning-tree mode	By default, STP, RSTP, and MSTP are disabled

TABLE 25 Default STP configuration (continued)

Parameter	Default setting
Bridge priority	32768
Bridge forward delay	15 seconds
Bridge maximum aging time	20 seconds
Error disable timeout timer	Disabled
Error disable timeout interval	300 seconds
Port-channel path cost	Standard
Bridge hello time	2 seconds

The following table lists the switch defaults for the interface-specific configuration.

TABLE 26 Default interface specific configuration

Parameter	Default setting
Spanning tree	Enabled on the interface
Automatic edge detection	Disabled
Path cost	2000
Edge port	Disabled
Guard root	Disabled
Hello time	2 seconds
Link type	Point-to-point
Portfast	Disabled
Port priority	128
BPDU restriction	Restriction is disabled.

STP features

The following sections discuss root guard, BPDU guard, and PortFast.

Root guard

Root guard can be used to predetermine a root bridge location and prevent rogue or unwanted switches from becoming the root bridge.

At times it is necessary to protect the root bridge from malicious attack or even unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge, causing severe bottlenecks in the data path. These types of mistakes or attacks can be avoided by configuring root guard on ports of the root bridge.

The root guard feature provides a way to enforce the root bridge placement in the network and allows STP and its variants to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires. Errors are triggered if any change from the root bridge placement is detected.

When root guard is enabled on a port, it keeps the port in designated FORWARDING state. If the port receives a superior BPDU, which is a root guard violation, it sets the port into a DISCARDING state and triggers a Syslog message and an SNMP trap. No further traffic will be forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or wrongly configured STP or RSTP bridges.

Root guard should be configured on all ports where the root bridge should not appear. In this way, the core bridged network can be cut off from the user network by establishing a protective perimeter around it.

Once the port stops receiving superior BPDUs, root guard automatically sets the port back to a FORWARDING state after the timeout period has expired.

BPDU guard

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow.

In a valid configuration, edge port-configured interfaces do not receive BPDUs. If an edge port-configured interface receives a BPDU, an invalid configuration exists, such as the connection of an unauthorized device. The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service.

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Extreme device port to which the end station is connected. The STP BPDU guard shuts down the port and puts it into an "error disabled" state. This disables the connected device's ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a message is displayed to warn the network administrator of an invalid configuration.

The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service with the **no shutdown** command if error disable recovery is not enabled by enabling the **errdisable-timeout** command. The interface can also be automatically configured to be enabled after a timeout. However, if the offending BPDUs are still being received, the port is disabled again.

Expected behavior in an interface context

When BPDU Guard is enabled on an interface, the device is expected to put the interface in Error Disabled state when BPDU is received on the port when edge-port and BPDU guard is enabled on the switch interface. When the port ceases to receive the BPDUs, it does not automatically switch to edge port mode, you must configure **error disable timeout** or **no shutdown** on the port to move the port back into edge port mode.

Error disable recovery

A port is placed into an error-disabled state when:

- A BPDU guard violation or loop detection violation occurs
- The number of inError packets exceeds the configured threshold
- An EFM-OAM enabled interface receives a critical event from the remote device (functionally equivalent to a disable state)

Once in an error disable state, the port remains in that state until it is re-enabled automatically or manually.

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, you can specify the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds. By default, the timeout feature is disabled.

PortFast

PortFast allows an interface to transition quickly to the forwarding state.

Consider the following when configuring PortFast:

- Do not enable PortFast on ports that connect to other devices.
- PortFast only needs to be enabled on ports that connect to workstations or PCs. Repeat this configuration for every port connected to workstations or PCs.
- Enabling PortFast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.
- If BPDUs are received on a PortFast- enabled interface, the interface loses the edge port status unless it receives a **shutdown**/**no shutdown** command.
- PortFast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

STP parameters

The following section discusses bridge parameters.

Bridge parameters

These parameters are set in STP, RSTP, MSTP, PVST+, and R-PVST+.

Bridge priority

Use this parameter to specify the priority of a device and to determine the root bridge.

Each device has a unique bridge identifier called the bridge ID. The bridge ID is an 8 byte value that is composed of two fields: a 2 B bridge priority field and the 6 B MAC address field. The value for the bridge priority ranges from 0 to 61440 in increments of 4096. The default value for the bridge priority is 32768. You use the **bridge-priority** command to set the appropriate values to designate a device as the root bridge or root device. A default bridge ID may appear as 32768.768e.f805.5800. If the bridge priorities are equal, the device with the lowest MAC address is elected the root.

After you decide what device to designate as the root, you set the appropriate device bridge priorities. The device with the lowest bridge priority becomes the root device. When a device has a bridge priority that is lower than that of all the other devices, it is automatically selected as the root.

The root device should be centrally located and not in a "disruptive" location. Backbone devices typically serve as the root because they usually do not connect to end stations. All other decisions in the network, such as which port to block and which port to put in forwarding mode, are made from the perspective of the root device.

You may also specify the bridge priority for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge Protocol data units (BPDUs) carry information between devices. All the devices in the Layer 2 network, participating in any variety of STP, gather information on other devices in the network through an exchange of BPDUs. As the result of exchange of the BPDUs, the device with the lowest bridge ID is elected as the root bridge

When setting the bridge forward delay, bridge maximum aging time, and the hello time parameters keep in mind that the following relationship should be kept:

```
(2 × (forward-delay − 1)) ≥ max-age ≥ (2 ×(hello-time + 1))
```

Bridge forward delay

The bridge forward delay parameter specifies how long an interface remains in the listening and learning states before the interface begins forwarding all spanning tree instances. The valid range is from 4 through 30 seconds. The default is 15 seconds.

Additionally, you may specify the forward delay for a specific VLAN. If the VLAN parameter is not provided, the bridge forward delay value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge maximum aging time

You can use this setting to configure the maximum length of time that passes before an interface saves its BPDU configuration information.

Keeping with the inequality shown above, when configuring the maximum aging time, you must set the value greater than the hello time. The range of values is 6 through 40 seconds while he default is 20 seconds.

You may specify the maximum aging for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge hello time

You can use this parameter to set how often the device interface broadcasts hello BPDUs to other devices.

Use the **hello-time** command to configure the bridge hello time. The range is from 1 through 10 seconds. The default is 2 seconds.

You may also specify the hello time for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Error disable timeout parameter

Configure this parameter to enable a timer that brings a port out of the disabled state.

These parameters are be set in STP, RSTP, MSTP, PVST+, and R-PVST+.

When the STP BPDU guard disables a port, the port remains in the disabled state unless the port is enabled manually. The parameter specifies the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds.

By default, the timeout feature is disabled.

Port-channel path cost parameter

You configure this parameter to specify the port channel path cost.

This parameter can be set in STP, RSTP, MSTP, PVST+, and R-PVST+ mode.

There are two path cost options:

- Custom - Specifies that the path cost changes according to the port channel bandwidth.
- Standard - Specifies that the path cost does not change according to the port channel bandwidth.

The default port cost is standard.

Configuring STP

The following section discusses configuring STP.

Enabling and configuring STP globally

Follow these steps to enable or disable STP and configure STP parameters.

You can enable STP or STP with one or more parameters enabled.

The parameters can be configured individually by:

1. Entering the commands in steps 1 and 2
2. Running the relevant parameter command
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable STP globally.

   ```
   device(config)# protocol spanning-tree stp
   ```

   A spanning tree can be disabled by entering the **no protocol spanning-tree stp** command.

3. Describe or name the STP.

   ```
   device(config-stp)# description stp1
   ```

   A description is not required.

4. Specify the bridge priority.

   ```
   device(config-stp)# bridge-priority 4096
   ```

   The bridge with the lowest priority number (highest priority) is designated the root bridge . The range of values is 0 through 61440; values can be set only in increments of 4096. The default priority is 32678.

5. Specify the bridge forward delay.

   ```
   device(config-stp)# forward-delay 20
   ```

   The forward delay specifies how long an interface remains in the listening and learning states before it begins forwarding all spanning tree instances. The valid range is from 4 through 30 seconds. The default is 15 seconds.

6. Configure the maximum aging time.

   ```
   device(config-stp)# max-age 25
   ```

   This parameter controls the maximum length of time that passes before an interface saves its BPDU configuration information. You must set the maximum age to be greater than the hello time. The range is 6 through 40 seconds. The default is 20 seconds.

7. Configure the maximum hello time.

```
device(config-stp)# hello-time 8
```

   The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The default is 2 seconds while the range is from 1 through 10 seconds.

8. Enable the error disable timeout timer.

```
device(config-stp)# error-disable-timeout enable
```

   This parameter enables a timer that brings a port out of the disabled state. By default, the timeout feature is disabled.

9. Set the error disable timeout timer.

```
device(config-stp)# error-disable-timeout interval 60
```

   When enabled the default is 300 seconds and the range is from 10 through 1000000 seconds.

10. Configure the port channel path cost.

```
device(config-stp)# port-channel path-cost custom
```

   Specifying **custom** means the path cost changes according to the port channel's bandwidth.

11. Return to privileged EXEC mode.

```
device(config-stp)# end
```

12. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: Spanning Tree Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 -------------------------------------------------------------
 Eth 0/2      DES   FWD  2000       128   P2P            No
 Eth 0/20     DIS   DIS  20000000   128   P2P            No
 Eth 0/25     DIS   DIS  20000000   128   P2P            No
 Eth 0/30     DIS   DIS  20000000   128   P2P            No
 Eth 0/31     DIS   DIS  2000000    128   P2P            No
```

   Observe that the settings comply with the formula set out in the STP parameter configuration section, as:

   `(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))`

   Or in this case `38 ≥ 25 ≥ 18`.

13. Save the configuration.

```
device# copy running-config startup-config
```

STP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# description stpForInterface
device(config-stp)# bridge-priority 4096
device(config-stp)# forward-delay 20
device(config-stp)# max-age 25
device(config-stp)# hello-time 8
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config-stp)# port-channel path-cost custom
device(config-stp)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Enabling and configuring STP on an interface

Follow these steps to enable STP and STP features on an interface.

Globally enable STP and STP parameters.

The parameters can be configured individually by:

1. Entering the commands in steps 1-3
2. Running the relevant parameter command
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface ethernet 0/20
   ```

3. Enable the interface.

   ```
   device(conf-if-eth-0/20)# no shutdown
   ```

4. Configure the path cost for spanning tree calculations on the interface.

   ```
   device(conf-if-eth-0/20)# spanning-tree cost 10000
   ```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

5. Enable BPDU guard on the interface.

   ```
   device(conf-if-eth-0/20)# spanning-tree port-fast bpdu-guard
   ```

   BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

6. Configure Root Guard on the interface.

```
device(conf-if-eth-0/20)# spanning-tree guard root
```

   Root Guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.

7. Specify an interface link-type.

```
device(conf-if-eth-0/20)# spanning-tree link-type point-to-point
```

   Specifying a point-to-point link enables rapid spanning tree transitions to the forwarding state. Specifying a shared link disables spanning tree rapid transitions. The default setting is point-to-point.

8. Specify port priority to influence the selection of root or designated ports.

```
device(conf-if-eth-0/20)# spanning-tree priority 64
```

   The range is from 0 through 240 in increments of 16. The default value is 128.

9. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: Spanning Tree Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 ------------------------------------------------------------
 Eth 0/2      DES   FWD  2000       128   P2P            No
 Eth 0/20     DES   FWD  1000        64   P2P            No
 Eth 0/25     DIS   DIS  20000000   128   P2P            No
 Eth 0/30     DIS   DIS  20000000   128   P2P            No
 Eth 0/31     DIS   DIS  2000000    128   P2P            No
```

   NOTE

   Observe that the settings comply with the formula set out in the STP parameters section, as:
   `(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))`
   or in this case: `38 ≥ 25 ≥ 18`

```
device# show running-config interface ethernet 0/20
interface ethernet 0/20
switchport
switchport mode access
switchport access val 1
spanning-tree cost 1000
spanning-tree guard root
spanning-tree link-type point-to-point
spanning-tree portfast bpdu-guard
spanning-tree priority 64
```

10. Save the settings by copying the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

STP on an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/20
device(conf-if-eth-0/20)# no shutdown
device(conf-if-eth-0/20)# spanning-tree cost 10000
device(conf-if-eth-0/20)# spanning-tree port-fast bpdu-guard
device(conf-if-eth-0/20)# spanning-tree guard root
device(conf-if-eth-0/20)# spanning-tree link-type point-to-point
device(conf-if-eth-0/20)# spanning-tree priority 64
device(conf-if-eth-0/20)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Configuring basic STP parameters

Follow this example to configure basic STP behavior.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable STP globally

   ```
   device(config)# protocol spanning-tree stp
   ```

3. Name the STP.

   ```
   device(config-stp)# description stp1
   ```

4. Designate the root switch.

   ```
   device(conf-stp)# bridge-priority 28672
   ```

   The priority values can be set only in increments of 4096. The range is 0 through 61440.

5. Specify the bridge forward delay.

   ```
   device(config-stp)# forward-delay 20
   ```

6. Configure the maximum aging time.

   ```
   device(config-stp)# max-age 25
   ```

7. Configure the maximum hello time.

   ```
   device(config-stp)# hello-time 8
   ```

8. Enable the error disable timeout timer.

   ```
   device(config-stp)# error-disable-timeout enable
   ```

9. Set the error disable timeout timer interval.

   ```
   device(config-stp)# error-disable-timeout interval 60
   ```

10. Enable port fast on switch ports.
    a) Configure port fast on Ethernet port 0/1.

```
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree portfast
device(conf-if-eth-0/1)# exit
```

    Spanning trees are automatically enabled on switch ports.

    b) Configure port fast on Ethernet port 0/2.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree portfast
device(conf-if-eth-0/2)# exit
```

    c) Repeat these commands for every port connected to workstations or PCs.

```
device(config)# interface ethernet ...
```

11. Specify port priorities to influence the selection of the root and designated ports.

```
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree priority 1
device(conf-if-eth-0/1)# exit
```

12. Enable the guard root feature.

```
device(config)# interface ethernet 0/12
device(conf-if-eth-0/12)# no shutdown
device(conf-if-eth-0/12)# spanning-tree guard root
```

    Root guard lets the device top participate in the STP but only when the device does not attempt to become the root.

13. Return to privileged exec mode.

```
device(conf-if-eth-0/12)# end
```

14. Verify the configuration.

```
device# show spanning-tree brief
Spanning-tree Mode: Spanning Tree Protocol
Root ID Priority 4096
Address 768e.f805.5800
Hello Time 8, Max Age 25, Forward Delay 20
Bridge ID Priority 4096
Address 768e.f805.5800
Hello Time 8, Max Age 25, Forward Delay 20
Interface     Role     Sts     Cost     Prio     Link-type     Edge
-------------------------------------------------------------------
Eth 0/1       DES      FWD     2000     128      P2P           No
Eth 0/2       DES      FWD     2000     128      P2P           No
Eth 0/12      DES      FWD     2000     128      P2P           No
```

    Observe that the settings comply with the formula set out in the STP parameter configuration section, as:

    `(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))`

    or in this case `38 ≥ 25 ≥ 18`.

15. Save the configuration.

```
device# copy running-config startup-config
```

Basic STP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# description stp1
device(conf-stp)# bridge-priority 28672
device(config-stp)# forward-delay 20
device(config-stp)# max-age 25
device(config-stp)# hello-time 8
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree portfast
device(conf-if-eth-0/1)# exit
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree portfast
device(conf-if-eth-0/2)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree priority 1
device(conf-if-eth-0/1)# exit
device(config)# interface ethernet 0/12
device(conf-if-eth-0/12)# no shutdown
device(conf-if-eth-0/12)# spanning-tree guard root
device(conf-if-eth-0/12)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Re-enabling an error-disabled port automatically

Enable a port to automatically recover from the error-disabled state after the expiration of an error recovery timer.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter STP configuration mode.

   ```
   device(config)# protocol spanning-tree stp
   ```

3. Enable the error-disable-timeout timer.

   ```
   device(config-stp)# error-disable-timeout enable
   ```

4. Set an interval after which port shall be enabled.

   ```
   device(config-stp)# error-disable-timeout interval 60
   ```

   The interval range is from 0 to 1000000 seconds, the default is 300 seconds.

5. Return to privileged EXEC mode.

   ```
   device(config-stp)# end
   ```

6. Verify the configuration.

```
device# show spanning-tree
 Spanning-tree Mode: Spanning Tree Protocol

 Root Id: 8000.768e.f805.5800 (self)
 Bridge Id: 8000.768e.f805.5800

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: enabled
 Bpdu-guard errdisable timeout interval: 60 sec
```

Automatically re-enable an error-disabled port configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config-stp)# end
device# show spanning-tree
```

Clearing spanning tree counters

Follow these steps to clear spanning tree counters on all interfaces or on the specified interface.

1. Clear spanning tree counters on all interfaces.

```
device# clear spanning-tree counter
```

2. Clear spanning tree counters on a specified Ethernet interface.

```
device# clear spanning-tree counter interface ethernet 0/3
```

3. Clear spanning tree counters on a specified port channel interface.

```
device# clear spanning-tree counter interface port-channel 12
```

   Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

Follow these steps to restart the protocol migration process.

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

```
device# clear spanning-tree detected-protocols
```

2. Restart the spanning tree migration process on a specific Ethernet interface.

```
device# clear spanning-tree detected-protocols interface ethernet 0/3
```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

   Port channel interface numbers range from 1 through 64.

Shutting down STP

Follow these steps to o shut down STP either globally, on a specific interface, or a specific VLAN.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down STP.
   - Shut down STP globally and return to privileged EXEC mode.

```
device(config)# protocol spanning-tree stp
device(config-stp)# shutdown
device(config-stp)# end
```

   - Shut down STP on a specific interface and return to privileged EXEC mode.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree shutdown
device(conf-if-eth-0/2)# end
```

   - Shut down STP on a specific VLAN and return to privileged EXEC mode.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
```

3. Verify the configuration.

```
device# show spanning-tree
device#
```

4. Save the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

Shut down STP configuration example

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-stp)# end
device# show spanning-tree
device# copy running-config startup-config
```

NOTE

Shutting down STP on a VLAN is used in this example.

802.1w Rapid Spanning Tree Protocol

- Rapid Spanning Tree Protocol overview 117
- Configuring RSTP..... 118

Rapid Spanning Tree Protocol overview

The RSTP is a way to provide rapid traffic reconvergence for point-to-point links within a few milliseconds (< 500 milliseconds), following the failure of a bridge or bridge port.

The STP (802.1d) standard was designed at a time when recovering connectivity after an outage within a minute or so was considered adequate performance. With the advent of Layer 3 switching in LAN environments, bridging competes with routed solutions where protocols such as OSPF are able to provide an alternate path in less time.

The RSTP can be seen as evolution of STP standard. It provides rapid convergence of connectivity following the failure of bridge, a bridge port or a LAN. It provides rapid convergence of edge ports, new root ports and port connected through point-to-point links. The port, which qualifies for fast convergence, is derived from the duplex mode of a port. A port operating in full-duplex will be assumed to be point-to-point, while a half-duplex port will be considered as a shared port by default. This automatic setting can be overridden by explicit configuration.

RSTP is designed to be compatible and interoperate with the STP. However, the benefit of the RSTP fast convergence is lost when interacting with legacy STP (802.1d) bridges since the RSTP downgrades itself to the STP when it detects a connection to a legacy bridge.

The states for every Layer 2 interface running the RSTP are as follows:

State	Action
Learning	The interface prepares to participate in frame forwarding.
Forwarding	The interface forwards frames.
Discarding	The interface discards frames. Ports in the discarding state do not take part in the active topology and do not learn MAC addresses.

NOTE
The STP disabled, blocking, and listening states are merged into the RSTP discarding state.

The RSTP port roles for the interface are also different. The RSTP differentiates explicitly between the state of the port and the role it plays in the topology. The RSTP uses the root port and designated port roles defined im the STP, but splits the blocked port role into backup port and alternate port roles:

Backup port	Provides a backup for the designated port and can only exist where two or more ports of the switch are connected to the same LAN; the LAN where the bridge serves as a designated switch.
Alternate port	Serves as an alternate port for the root port providing a redundant path towards the root bridge.

Only the root port and the designated ports are part of the active topology; the alternate and backup ports do not participate in it. When the network is stable, the root and the designated ports are in the forwarding state, while the alternate and backup ports are in the discarding state. When there is a topology change, the new RSTP port roles allow a faster transition of an alternate port into the forwarding state.

For more information about spanning trees, see the introductory sections in the 802.1d Spanning Tree Protocol chapter.

RSTP parameters

The parameters you would normally set when you configure STP are applicable to RSTP. Before you configure RSTP see the STP parameters sections for descriptions of the bridge parameters. the error disable timeout parameter and the port channel path cost parameter.

There is one parameter that can be configured in RSTP that is not available in STP; the transmit hold count. This parameter configures the BPDU burst size by specifying the maximum number of BPDUs transmitted per second for before pausing for 1 second. The range is 1 through 10 while the default is 6. See the section Enabling RSTP and configuring RSTP parameters for the procedure to configure this parameter.

The edge port and auto edge features can be enabled in RSTP as well. See the section Edge port and automatic edge detection and the section Configuring RSTP on an interface for descriptions of these features and how they are configured.

Edge port and automatic edge detection

Configuring the edge port feature makes a port transition directly from initialization to the forwarding state, skipping the listening and learning states.

From an interface, you can configure a device to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

Follow these guidelines to configure a port as an edge port:

- When edge port is enabled, the port still participates in a spanning tree.
- A port can become an edge port if no BPDU is received.
- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

  NOTE
  If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

Configuring RSTP

Enabling and configuring RSTP globally

Follow these steps to enable and configure RSTP.

See the section STP parameters for parameters applicable to all STP variants.

You can enable RSTP or RSTP with one or more parameters enabled. The parameters can be enabled or changed individually by entering the commands in steps 1 and 2, running the parameter command, verifying the result, and then saving the configuration.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable RSTP.

```
device(config)# protocol spanning-tree rstp
```

   You can shut down RSTP by entering the **shutdown** command.

3. Designate the root device.

```
device(conf-rstp)# bridge-priority 28582
```

   The range is 0 through 61440 and the priority values can be set only in increments of 4096.

   You can shut down RSTP by entering the **shutdown** command when in RSTP configuration mode.

4. Configure the bridge forward delay value.

```
device(conf-rstp)# forward-delay 15
```

5. Configure the bridge maximum aging time value.

```
device(conf-rstp)# max-age 20
```

6. Enable the error disable timeout timer.
   a) Enable the timer.

```
device(conf-rstp)# error-disable-timeout enable
```

   b) Configure the error disable timeout interval value.

```
device(conf-rstp)# error-disable-timeout interval 60
```

7. Configure the port-channel path cost.

```
device(conf-rstp)# port-channel path-cost custom
```

8. Configure the bridge hello-time value.

```
device(conf-rstp)# hello-time 2
```

9. Specify the transmit hold count.

```
device(config-rstp)# transmit-holdcount 5
```

   This command configures the maximum number of BPDUs transmitted per second.

10. Return to privileged exec mode.

```
device(conf-rstp)# end
```

11. Verify the configuration

```
device# show spanning-tree

 Spanning-tree Mode: Rapid Spanning Tree Protocol

 Root Id: 8000.01e0.5200.0180 (self)
 Bridge Id: 8000.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: enabled
 Bpdu-guard errdisable timeout interval: 60 sec
```

    Observe that the settings comply with the formula set out in the STP parameters section, as:

    (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

    or in this case: 28 ≥ 20 ≥ 6.

12. Save the configuration.

```
device# copy running-config startup-config
```

Enabling RSTP and configuring RSTP parameters example

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(conf-rstp)# bridge-priority 28582
device(conf-rstp)# forward-delay 20
device(conf-rstp)# max-age 25
device(conf-rstp)# error-disable-timeout enable
device(conf-rstp)# error-disable-timeout interval 60
device(conf-rstp)# port-channel path-cost custom
device(conf-rstp)# hello-time 5 forward-delay 16 max-age 21
device(conf-rstp)# transmit-holdcount 5
device(conf-rstp# end
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring RSTP on an interface

Follow these steps to configure RSTP on an Ethernet interface.

You can configure the parameters individually on an interface by doing the following:

1. Entering the commands in Steps 1 through 3.
2. Specifying additional parameters, as appropriate.
3. Verifying the result.
4. Saving the configuration.

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface subtype configuration mode.

```
device(config)# interface ethernet 0/10
```

3. Enable the interface.

```
device(conf-if-eth-0/10)# no shutdown
```

To disable the spanning tree on the interface you use the **spanning-tree shutdown** command.

4. Specify the port priority on the interface.

```
device(conf-if-eth-0/10)# spanning-tree priority 128
```

The range is from 0 through 240 in increments of 16. The default value is 128.

5. Specify the path cost on the interface.

```
device(conf-if-eth-0/10)# spanning-tree cost 20000000
```

The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

6. Enable edge port.

```
device(conf-if-eth-0/10)# spanning-tree edgeport
```

If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

7. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/10)# spanning-tree edgeport bpdu-guard
```

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

8. Enable automatic edge detection on the interface.

```
device(conf-if-eth-0/10)# spanning-tree autoedge
```

You use this command to automatically identify the edge port. A port becomes an edge port if it receives no BPDUs. By default, automatic edge detection is disabled.

9. Enable root guard on the interface.

```
device(conf-if-eth-0/10)# spanning-tree guard root
```

Root guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.

10. Specify a link type on the interface.

```
device(conf-if-eth-0/10)# spanning-tree link-type point-to-point
```

NOTE
The link type is explicitly configured as **point-to-point** rather than **shared**.

11. Return to privileged EXEC mode.

```
device(conf-if-eth-0/10)# end
```

12. Verify the configuration.

```
device# show spanning-tree

 Spanning-tree Mode: Rapid Spanning Tree Protocol

 Root Id: 8000.01e0.5200.0180 (self)
 Bridge Id: 8000.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: enabled
 Bpdu-guard errdisable timeout interval: 60 sec


 Port Eth 0/10 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Spanning Tree Protocol - Received None - Sent STP
    Edgeport: on; AutoEdge: yes; AdminEdge: no; EdgeDelay: 3 sec
    Configured Root guard: on; Operational Root guard: on
    Bpdu-guard: on
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

Interface   Role     Sts     Cost       Prio     Link-type     Edge
-------------------------------------------------------------------
Eth 0/10    DES      FWD     20000000   128      P2P           No
```

The **forward-delay**, **hello-time**, and **max-age** parameters are set globally, not on the interface.

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

13. Save the configuration.

```
device# copy running-config startup-config
```

RSTP on an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/10
device(conf-if-eth-0/10)# no spanning-tree shutdown
device(conf-if-eth-0/10)# spanning-tree priority 128
device(conf-if-eth-0/10)# spanning-tree cost 20000000
device(conf-if-eth-0/10)# spanning-tree edgeport
device(conf-if-eth-0/10)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-0/10)# spanning-tree autoedge
device(conf-if-eth-0/10)# spanning-tree guard root
device(conf-if-eth-0/10)# spanning-tree link-type point-to-point
device(conf-if-eth-0/10)# end
device# show spanning-tree
device# copy running-config startup-config
```

Configuring basic RSTP parameters

Follow these steps to configure basic RSTP parameters.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable RSTP.

```
device(config)# protocol spanning-tree rstp
```

3. Designate the root device.

```
device(conf-rstp)# bridge-priority 28582
```

4. Enable the error disable timeout timer value.

```
device(conf-rstp)# error-disable-timeout enable
```

5. Configure the error-disable-timeout interval value.

```
device(conf-rstp)# error-disable-timeout interval 60
```

6. Enable edge port on switch ports.
   a) Enter interface subtype configuration mode for the switchport.

```
device(conf-rstp)# interface ethernet 0/10
```

   b) Enable edge port.

```
device(conf-if-eth-0/10)# spanning-tree edge-port
```

   c) Return to global configuration mode.

```
device(conf-if-eth-0/10)# exit
```

   d) Repeat the above steps for all ports that connect to a workstation or PC.
7. Specify port priorities.
   a) Enter interface subtype configuration mode.

```
device(config)# interface ethernet 0/11
```

   b) Configure the port priority.

```
device(conf-if-eth-0/11)# spanning-tree priority 1
```

   c) Return to global configuration mode.

```
device(conf-if-eth-0/11)# exit
```

8. Enable the guard root feature.
   a) Enter interface configuration mode.

```
device(config)# interface ethernet 0/1
```

   b) Configure the port priority.

```
device(conf-if-eth-0/1)# spanning-tree guard root
```

   c) Return to privileged EXEC mode.

```
device(conf-if-eth-0/1)# exit
```

9. Verify the configuration.

```
device# show spanning-tree

Spanning-tree Mode: Rapid Spanning Tree Protocol

Root Id: 4096.01e0.5200.0180 (self)
Bridge Id: 4096.01e0.5200.0180
Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
Number of topology change(s): 0
Bpdu-guard errdisable timeout: disabled
Bpdu-guard errdisable timeout interval: 300 sec
switch# show spanning-tree brief
Spanning-tree Mode: Rapid Spanning Tree Protocol
Root ID Priority 4096
Address 768e.f805.5800
Hello Time 2, Max Age 20, Forward Delay 15
Bridge ID Priority 4096
Address 768e.f805.5800

Interface   Role     Sts     Cost     Prio     Link-type     Edge
-------------------------------------------------------------------
Eth 0/1     DES      FWD     2000     128      P2P           No
Eth 0/10    DES      FWD     2000     128      P2P           No
Eth 0/11    DES      FWD     2000     128      P2P           No
```

   Observe that the settings comply with the formula set out in the STP parameters section, as follows:

   (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

   or in this case: 28 ≥ 20 ≥ 6.

10. Save the configuration.

```
device# copy running-config startup-config
```

Basic RSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(conf-rstp)# bridge-priority 28582
device(conf-rstp)# error-disable-timeout enable
device(conf-rstp)# error-disable-timeout interval 60
device(conf-rstp)# interface etherne 0/10
device(conf-if-eth-0/10)# spanning-tree edge-port
device(conf-if-eth-0/10)# exit
device(config)# interface ethernet 0/11
device(conf-if-eth-0/11)# spanning-tree priority 1
device(conf-if-eth-0/11)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree guard root
device(conf-if-eth-0/1)# exit
device# show spanning-tree
device# copy running-config startup-config
```

Clearing spanning tree counters

Follow these steps to clear spanning tree counters on all interfaces or on the specified interface.

1. Clear spanning tree counters on all interfaces.

   ```
   device# clear spanning-tree counter
   ```

2. Clear spanning tree counters on a specified Ethernet interface.

   ```
   device# clear spanning-tree counter interface ethernet 0/3
   ```

3. Clear spanning tree counters on a specified port channel interface.

   ```
   device# clear spanning-tree counter interface port-channel 12
   ```

   Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

Follow these steps to restart the protocol migration process.

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

   ```
   device# clear spanning-tree detected-protocols
   ```

2. Restart the spanning tree migration process on a specific Ethernet interface.

   ```
   device# clear spanning-tree detected-protocols interface ethernet 0/3
   ```

3. Restart the spanning tree migration process on a specific port channel interface.

   ```
   device# clear spanning-tree detected-protocols port-channel 12
   ```

   Port channel interface numbers range from 1 through 64.

Shutting down RSTP

Follow these steps to shut down RSTP either globally or on a specific interface.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down RSTP.
   - Shut down STP globally and return to privileged EXEC mode.

```
device(config)# protocol spanning-tree rstp
device(conf-rstp)# shutdown
device(conf-rstp)# end
```

   - Shut down RSTP on a specific interface and return to privileged EXEC mode.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree shutdown
device(conf-if-eth-0/2)# end
```

   - Shut down RSTP on a specific VLAN and return to privileged EXEC mode.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
```

3. Verify the configuration.

```
device# show spanning-tree
device#
```

4. Save the configuration.

```
device# copy running-config startup-config
```

Per-VLAN Spanning Tree+ and Rapid Per-VLAN Spanning Tree+

- PVST+ and R-PVST+ overview 127
- Configuring PVST+ and R-PVST+ 132

PVST+ and R-PVST+ overview

The Per-VLAN Spanning Tree Plus (PVST+) protocol runs a spanning tree instance for each VLAN in the network. The version of PVST+ that uses the RSTP state machine is called Rapid-PVST Plus (R-PVST+). R-PVST+ has one instance of spanning tree for each VLAN on the device.

Both the STP and the RSTP build a single logical topology. A typical network has multiple VLANs. A single logical topology does not efficiently utilize the availability of redundant paths for multiple VLANs. A single logical topology does not efficiently utilize the availability of redundant paths for multiple VLANs. If a port is set to the blocked state or the discarding state for one VLAN (under the STP or the RSTP), it is the same for all other VLANs. PVST+ builds on the STP on each VLAN, and R-PVST+ builds on the RSTP on each VLAN.

PVST+ R-PVST+ provide interoperability with Cisco PVST and R-PVST and other vendor switches which implement Cisco PVST or R-PVST. the PVST+ and R-PVST+ implementations are extensions to PVST and R-PVST, which can interoperate with an STP topology, including MSTP (CIST), on Extreme and other vendor devices sending untagged IEEE BPDUs.

PVST+ and R-PVST+ guidelines and restrictions

Consider the following when configuring PVST+ and R-PVST+:

- Extreme supports PVST+ and R-PVST+ only. The PVST and R-PVST protocols are proprietary to Cisco and are not supported.
- A port native VLAN is the native VLAN ID associated with a trunk port on an Extreme switch. This VLAN ID is associated with all untagged packets on the port. The default native VLAN ID for a trunk port is 1.
- IEEE compliant switches run just one instance of STP protocol shared by all VLANs, creating a Mono Spanning Tree (MST). A group of such switches running a single spanning tree forms an MST region.
- You can configure up to 128 PVST+ or R-PVST+ instances. If you have more than 128 VLANs configured on the switch and enable PVST then the first 128 VLANs are PVST/+ or R-PVST+ enabled.
- In PVST/+ or R-PVST+ mode, when you are connected to a Cisco or MLX switch, the Cisco proprietary MAC address to which the BPDUs are sent/processed must be explicitly configured on a per-port basis.
- In PVST/+ or R-PVST+ mode, when you connect to a Cisco switch using a trunk port, the Extreme switch must have a native VLAN configured on the trunk port (same configuration as on the other side).
- A Common Spanning Tree (CST) is the single spanning tree instance used by Extreme switches to interoperate with 802.1q bridges. This spanning tree instance stretches across the entire network domain (including PVST, PVST+ and 802.1q regions). It is associated with VLAN 1 on the Extreme switch.
- In order to interact with STP and IEEE 802.1q trunk, PVST evolved to PVST+ to interoperate with STP topology by STP BPDU on the native or default VLAN.
- A group of switches running PVST+ is called a PVST+ region.

For more information about spanning trees, see the introductory sections in the Spanning Tree Protocol chapter.

PVST+ and R-PVST+ parameters

The parameters you would normally set when you configure STP are applicable to PVST+ and R-PVST+. Before you configure PVST+ or R-PVST+ parameters see the sections in the Standing Tree Protocol chapter explaining bridge parameters, the error disable timeout parameter and the port channel path cost parameter.

There is one parameter that can be configured in R-PVST+ that is not available in STP or PVST+; the transmit hold count. This parameter configures the BPDU burst size by specifying the maximum number of BPDUs transmitted per second for before pausing for 1 second. The range is 1 through 10 while the default is 6. See the section Configuring R-PVST+ for the procedure to configure this parameter.

Bridge protocol data units in different VLANs

PVST+ uses the spanning tree instance for VLAN 1 to join the CST in the network to build the CST, PVST+ processes and sends standard IEEE Bridge protocol data units (BPDUs) on all the ports in VLAN 1 (access/trunk).

Across IEEE 802.1q trunks, Extreme switches run PVST+. The goal is to interoperate with standard IEEE STP (or RSTP or MSTP), while transparently tunneling PVST+ instance BPDUs across the MST region to potentially connect to other Extreme switches across the MST region.

On trunk ports that allow VLAN 1, PVST+ also sends PVST+ BPDUs to a Cisco-proprietary multicast MAC address (0100.0ccc.cccd) or Extreme-proprietary multicast MAC address (0304.0800.0700) depending on the configuration. By default, the PVST+ BPDUs are sent to Extreme-proprietary multicast MAC address on Extreme switches. These BPDUs are tunneled across an MST region. The PVST + BPDUs for VLAN 1 are only used for the purpose of consistency checks and that it is only the IEEE BPDUs that are used for building the VLAN 1 spanning tree. So in order to connect to the CST, it is necessary to allow VLAN 1 on all trunk ports.

For all other VLANs, PVST+ BPDUs are sent on a per-VLAN basis on the trunk ports. These BPDUs are tunneled across an MST region. Consequently, for all other VLANs, MST region appears as a logical hub. The spanning tree instances for each VLAN in one PVST+ region map directly to the corresponding instances in another PVST+ region and the spanning trees are calculated using the per-VLAN PVST+ BPDUs.

Similarly, when a PVST+ region connects to a MSTP region, from the point of view of MSTP region, the boundary bridge thinks it is connected to a standard IEEE compliant bridge sending STP BPDUs. So it joins the CIST of the MSTP region to the CST of the PVST+ region (corresponding to VLAN 1). The PVST+ BPDUs are tunneled transparently through the MSTP region. So from the Extreme bridge point of view, the MSTP region looks like a virtual hub for all VLANs except VLAN 1.

The PVST+ BPDUs are sent untagged for the native VLAN and tagged for all other VLANs on the trunk port.

On access ports, Extreme switches run classic version of IEEE STP/RSTP protocol, where the BPDUs are sent to the standard IEEE multicast address "0180.C200.0000". So if we connect a standard IEEE switch to an access port on the Extreme switch, the spanning tree instance (corresponding to the access VLAN on that port) of the Extreme switch is joined with the IEEE STP instance on the adjacent switch.

For introductory information about STP BPDUs, see the section BPDUs on page 102.

BPDU configuration notes

In order to build a spanning tree for the bridge topology, the bridges must exchange control frames. These frames are called Bridge Protocol data units (BPDU).

Topology Change Notification (TCN) BPDUs are used to inform other switches of port changes. TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

In PVST+, three types of TCN BPDUs are sent out depending on the type of the link. See Extreme PVST+ TCN BPDU headers/fields on page 131 and Cisco PVST TCN BPDU headers/fields on page 131.

- Standard IEEE TCN BPDU.
- Untagged TCN BPDU sent to the Cisco/Extreme proprietary MAC address.
- Tagged TCN BPDU sent to the Cisco/Extreme proprietary MAC address.

BPDU R-PVST+ header and field comparisons

These tables outline the differences between Brocade R-PVST+ BPDU and Cisco R-PVST+ BPDU header fields.

Brocade R-PVST+ BPDU headers/fields

Header/field	Standard IEEE STP/RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
Source Address (MAC SA)	6B	6B	6B
Destination Address (MAC DA)	0180C2.000000 (6B)	030408.000700 (6B)	030408.000700 (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
Source Service Access Point (SSAP)	42	AA 03	AA 03
Destination Service Access Point (DSAP)	42	AA	AA
Brocade Organizationally Unique Identifier (OUI)	-	02 04 08	02 04 08
PVST PID	-	01 0B	01 0B
Logical Link Control (LLC)	3B	+	+
SubNetwork Access Protocol (SNAP)	-	Yes (2B)	Yes (2B)
IEEE BPDU INFO	35B	35B	35B
Type, Length, Value (TLV) Pad Type Length VLAN ID	-	6B 00 (1B) 00 00 00 02 2B	6B 00 (1B) 00 00 00 02 2B

Cisco R-PVST+ BPDU headers/fields

Header/field	Standard IEEE STP/RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	01000C.CCCCCD (6B)	010002.CCCCCD (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03

Header/field	Standard IEEE STP/RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
DSAP	42	AA	AA
Cisco OUI	-	00 00 0C	00 00 0C
PVST PID	-	01 0B	01 0B
LLC	3B	+	+
SNAP	-	Yes	Yes
IEEE BPDU INFO	35B	35B	35B
TLV Pad Type Length VLAN ID	-	6B 00 (1B) 00 00 00 02 2B	6B 00 (1B) 00 00 00 02 2B

Sent BPDUs

On an 802.1q trunk, the PVST+ enabled switch sends the following BPDUs:

1.
2. If PVST+ is enabled on the untagged (native) VLAN of the port, an untagged SSTP BPDU is sent to the Brocade or Cisco MAC address on the native VLAN of the trunk. It is possible that the native VLAN on the Brocade or Cisco port is not VLAN 1. This BPDU is also forwarded on the native VLAN of the IEEE 802.1q switch just like any other frame sent to an unknown multicast address.
3.
4. A standard IEEE BPDU (802.1d) is also sent, corresponding to the information of VLAN 1 on the Brocade or Cisco switch. This BPDU is not sent if VLAN 1 is explicitly disabled on the trunk port.

The following table lists the types of BPDUs sent in case of different port types. The numbers in the third column are the VLAN instance for which these BPDUs are sent/processed.

TABLE 27 Types of BPDUs sent for different port types

Port Configuration	Brocade or Cisco - PVST(+)	VLAN instance
Access - VLAN 1	Standard IEEE BPDU (64B)	1
Access - VLAN 100	Standard IEEE BPDU (64B)	100
Trunk - Native VLAN 1 Allowed VLANs - 1, 100, 200	Standard IEEE BPDU (64B) Brocade or Cisco untagged BPDU (68B) Brocade or Cisco tagged BPDU (72B) Brocade or Cisco tagged BPDU (72B)	1 1 100 200
Trunk - Native VLAN 100 Allowed VLANs - 1, 100, 200	Standard IEEE BPDU (64B) Brocade or Cisco untagged BPDU (68B) Brocade or Cisco tagged BPDU (72B Brocade or Cisco tagged BPDU (72B)	1 100 1 200
Trunk - Native VLAN 100 Allowed VLANs - 100	Brocade or Cisco untagged BPDU (68B)	100

TABLE 27 Types of BPDUs sent for different port types (continued)

Port Configuration	Brocade or Cisco - PVST(+)	VLAN instance
Trunk - Native VLAN 100 Allowed VLANs - 100, 200	Brocade or Cisco untagged BPDU (68B) Brocade or Cisco tagged BPDU (72B)	100 200

TCN headers and fields

Since PVST+ is based on STP, and Rapid-PVST+ is based on RSTP, TCN BPDUs are sent only in PVST+ and not in Rapid-PVST+ mode.

For introductory information about STP BPDUs, see the section TCN BPDUs on page 103.

Brocade PVST+ TCN BPDU headers/fields

Header/field	Standard IEEE STP TCN BPDU (64B with padding)	PVST+ untagged TCN BPDU (64B with padding)	PVST+ tagged TCN BPDU (68B with padding)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	030408.000700 (6B)	030408.000700 ((6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	02 04 08	02 04 08
PVST PID	-	01 0B	01 0B
LLC	3B	8B	8B
SNAP	4B	Entire BPDU with type = TCN 35B	Entire BPDU with type = TCN 35B

Cisco PVST TCN BPDU headers/fields

Header/field	Standard IEEE STP TCN BPDU (64B padded)	PVST untagged TCN BPDU (64B padded)	PVST tagged TCN BPDU (68B padded)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	01000C.CCCCCD (6B)	01000C.CCCCCD (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	00 00 0C	00 00 0C
PVST PID	-	01 0B	01 0B
LLC	3B	8B	8B
SNAP	-	Yes	Yes
IEEE TCN BPDU INFO	4B	Entire BPDU with type = TCN 35B	Entire BPDU with type = TCN 35B

PortFast

PortFast allows an interface to transition quickly to the forwarding state.

Consider the following when configuring PortFast:

- Do not enable PortFast on ports that connect to other devices.
- PortFast only needs to be enabled on ports that connect to workstations or PCs. Repeat this configuration for every port connected to workstations or PCs.
- Enabling PortFast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.
- If BPDUs are received on a PortFast- enabled interface, the interface loses the edge port status unless it receives a **shutdown**/**no shutdown** command.
- PortFast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

Edge port and automatic edge detection

Configuring the edge port feature makes a port transition directly from initialization to the forwarding state, skipping the listening and learning states.

From an interface, you can configure a device to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

Follow these guidelines to configure a port as an edge port:

- When edge port is enabled, the port still participates in a spanning tree.
- A port can become an edge port if no BPDU is received.
- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

  NOTE
  If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

Configuring PVST+ and R-PVST+

Enabling and configuring PVST+ globally

Use this procedure to enable and set parameters for PVST+.

You can enable PVST+ with one or more parameters configured. The parameters can be configured or changed individually by entering the commands in steps 1 and 2, running the parameter command, verifying the result, and then saving the configuration.

For more information about spanning trees and spanning tree parameters, see the introductory sections in the Spanning Tree Protocol chapter.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable PVST+.

```
device(config)# protocol spanning-tree pvst
```

3. Configure the bridge priority for the common instance.

```
device(config-pvst)# bridge-priority 4096
```

Valid values range from 0 through 61440 in increments of 4096. Assigning a lower priority value indicates that the bridge might become root.

You can shut down PVST+ by entering the **shutdown** command when in PVST configuration mode.

4. Configure the forward delay parameter.

```
device(config-pvst)# forward-delay  11
```

5. Configure the hello time parameter.

```
device(config-pvst)# hello-time 2
```

6. Configure the maximum age parameter.

```
device(config-pvst)# max-age 7
```

7. Return to privileged exec mode.

```
device(config-pvst)# end
```

8. Verify the configuration.

```
device#  show spanning-tree brief
VLAN 1

 Spanning-tree Mode: PVST Protocol

      Root ID      Priority 4097
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 4097
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------

VLAN 100

 Spanning-tree Mode: PVST Protocol

      Root ID      Priority 4196
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 4196
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 20 ≥7 ≥ 6.

9. Save the configuration.

```
device# copy running-config startup-config
```

PVST+ configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(config-pvst)# bridge-priority 4096
device(config-pvst)# forward-delay 11
device(config-pvst)# hello-time 2
device(config-pvst)# max-age 7
device(config-pvst)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

For more information about configuring PVST+ parameters, see STP parameters on page 106. PVST+, R-PVST+, and other types of spanning trees share many tasks with STP.

Enabling and configuring PVST+ on an interface

Follow these steps to enable and configure PVST+ on an interface.

The ports and parameters can be configured individually on a system by:

1. Entering the commands in steps 1, and 2
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable PVST+.

```
device(config)# protocol spanning-tree pvst
```

3. Enter interface configuration mode.

```
device(config-pvst)# interface ethernet 0/3
```

4. Enable spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

5. Configure the interface link type.

```
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
```

6. Specify the port priority to influence the selection of root or designated ports.

```
device(conf-if-eth-0/3)# spanning-tree priority 64
```

   The range is from 0 through 240 in increments of 16. The default value is 128.

7. Configure the path cost for spanning tree calculations on the interface.

```
device(conf-if-eth-0/3)# spanning-tree cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

8. Configure the path cost for spanning tree calculations a specific VLAN.

```
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

9. Enable root guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree guard root
```

   Root guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.

10. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree port-fast bpdu-guard
```

   BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

11. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

12. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: PVST Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 -------------------------------------------------------------
 Eth 0/3      DES   FWD  200000     64    P2P            No
```

   Observe that the settings comply with the formula set out in the STP parameters section, as:

   (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

   or in this case :38 ≥ 25 ≥ 18.

13. Save the configuration.

```
device# copy running-config startup-config
```

PVST+ on an interface configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(conf-pvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
device(conf-if-eth-0/3)# spanning-tree priority 64
device(conf-if-eth-0/3)# spanning-tree cost 10000
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
device(conf-if-eth-0/3)# spanning-tree guard root
device(conf-if-eth-0/3)# spanning-tree port-fast bpdu-guard
device(conf-if-eth-0/3)# exit
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring PVST+ on a system

Follow the steps to configure PVST+ on a system.

The ports and parameters can be configured individually on a system by:

1. Entering the commands in steps 1, and 2
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable PVST+.

   ```
   device(config)# protocol spanning-tree pvst
   ```

3. Configure the bridge priority for the common instance.

   ```
   device(config-pvst)# bridge-priority 4096
   ```

   Valid values range from 0 through 61440 in multiples of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the forward delay parameter.

   ```
   device(config-pvst)# forward-delay  15
   ```

5. Configure the hello time parameter.

   ```
   device(config-pvst)# hello-time 2
   ```

6. Configure the maximum age parameter.

   ```
   device(config-pvst)# max-age 20
   ```

7. Add VLANs.
   a) Configure VLAN 100 with a priority of 0.

      ```
      device(config-pvst)# vlan 100 priority 0
      ```

      The bridge priority in configured in multiples of 4096.
   b) Configure VLAN 201 with a priority of 12288.

      ```
      device(config-pvst)# vlan 201 priority 12288
      ```

   c) Configure VLAN 301 with a priority of 20480.

      ```
      device(config-pvst)# vlan 301 priority 20480
      ```

8. Set the switching characteristics for interface 0/3.
   a) Enter interface configuration mode.

      ```
      device(config)# interface ethernet 0/3
      ```

   b) Set the switching characteristics of the interface.

      ```
      device(conf-if-eth-0/3)# switchport
      ```

   c) Set the interface mode to trunk.

      ```
      device(conf-if-eth-0/3)# switchport mode trunk
      ```

   d) Add VLAN 100 as a member VLAN.

      ```
      device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
      ```

   e) Add VLAN 201 as a member VLAN.

      ```
      device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
      ```

   f) Add VLAN 301 as a member VLAN.

      ```
      device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
      ```

   g) Enable spanning tree on the interface.

      ```
      device(conf-if-eth-0/3)# no spanning-tree shutdown
      ```

   h) Return to privileged EXEC mode.

      ```
      device(conf-if-eth-0/3)# exit
      ```

9. Set the switching characteristics for interface 0/4.
   a) Enter interface configuration mode.

```
device(config)# interface ethernet 0/4
```

   b) Set the switching characteristics of the interface.

```
device(conf-if-eth-0/4)# switchport
```

   c) Set the interface mode to trunk.

```
device(conf-if-eth-0/4)# switchport mode trunk
```

   d) Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
```

   e) Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
```

   f) Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
```

   g) Enable spanning tree on the interface.

```
device(conf-if-eth-0/4)# no spanning-tree shutdown
```

   h) Return to privileged EXEC mode.

```
device(conf-if-eth-0/4)# exit
```

10. To interoperate with switches other than VDX switches in PVST+ mode, you must configure the interface that is connected to that switch.
   a) Enter interface configuration mode for the port that interoperates with a VDX device.

```
device(config)# interface ethernet 0/12
```

   b) Specify the MAC address for the device.

```
device(conf-if-eth-0/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
```

   c) Enable spanning tree on the interface.

```
device(conf-if-eth-0/12)# no spanning-tree shutdown
```

   d) Return to privileged EXEC mode.

```
device(conf-if-eth-0/12)# end
```

11. Verify the configuration.

```
device# show spanning-tree

VLAN 1

 Spanning-tree Mode: PVST Protocol

 Root Id: 0001.01e0.5200.0180 (self)
 Bridge Id: 0001.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 100

 Spanning-tree Mode: PVST Protocol

 Root Id: 0064.01e0.5200.0180 (self)
 Bridge Id: 0064.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
```

```
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 201

 Spanning-tree Mode: PVST Protocol

 Root Id: 30c9.01e0.5200.0180 (self)
 Bridge Id: 30c9.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 301

 Spanning-tree Mode: PVST Protocol

 Root Id: 512d.01e0.5200.0180 (self)
 Bridge Id: 512d.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec
```

```
Port Et 0/3 enabled
   Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
   Designated Path Cost: 0
   Configured Path Cost: 20000000
   Designated Port Id: 0; Port Priority: 128
   Designated Bridge: 0000.0000.0000.0000
   Number of forward-transitions: 0
   Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
   Portfast: off
   Configured Root guard: off; Operational Root guard: off
   Bpdu-guard: off
   Link-type: point-to-point
   Received BPDUs: 0; Sent BPDUs: 0

Port Et 0/4 enabled
   Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
   Designated Path Cost: 0
   Configured Path Cost: 20000000
   Designated Port Id: 0; Port Priority: 128
   Designated Bridge: 0000.0000.0000.0000
   Number of forward-transitions: 0
   Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
   Portfast: off
   Configured Root guard: off; Operational Root guard: off
   Bpdu-guard: off
   Link-type: point-to-point
   Received BPDUs: 0; Sent BPDUs: 0
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

12. Save the configuration.

```
device# copy running-config startup-config
```

Enable PVST+ on a system configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(config-pvst)# bridge-priority 4096
device(config-pvst)# forward-delay  15
device(config-pvst)# hello-time 2
device(config-pvst)# max-age 20
device(config-pvst)# vlan 100 priority 0
device(config-pvst)# vlan 201 priority 12288
device(config-pvst)# vlan 301 priority 20480
device(config-pvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# switchport
device(conf-if-eth-0/3)# switchport mode trunk
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# exit
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# switchport
device(conf-if-eth-0/4)# switchport mode trunk
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/4)# no spanning-tree shutdown
device(conf-if-eth-0/4)# end
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring R-PVST+ globally

Use this procedure to enable the Rapid Per-VLAN Spanning Tree Protocol Plus (R-PVST+).

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable R-PVST+.

```
device(config)# protocol spanning-tree rpvst
```

3. Configure the bridge priority for the common instance.

```
device(config-rpvst)# bridge-priority 4096
```

   Valid priority values range from 0 through 61440 in multiples of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the forward delay parameter.

```
device(config-rpvst)# forward-delay  20
```

5. Configure the hello time parameter.

```
device(config-rpvst)# hello-time 22
```

6. Configure the maximum age parameter.

```
device(config-rpvst)# max-age 8
```

7. Set the transmit hold count for the bridge.

```
device(config-rpvst)# transmit-holdcount 9
```

   This command configures the maximum number of BPDUs transmitted per second before pausing for 1 second. The range is 1 through 10. The default is 6.

8. Return to privileged exec mode.

```
device(config-rpvst)# end
```

9. Verify the configuration.

```
device# show spanning-tree brief
VLAN 1

 Spanning-tree Mode: Rapid PVST Protocol

      Root ID      Priority 4096
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 32769
                   Address 01e0.5200.0180
                   Hello Time 8, Max Age 22, Forward Delay 20, Tx-HoldCount 9
                   Migrate Time 3 sec

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 -------------------------------------------------------------------
```

   Observe that the settings comply with the formula set out in the STP parameters section, as:

   (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

   or in this case: 20 ≥ 7 ≥ 6.

10. Save the configuration.

```
device# copy running-config startup-config
```

R-PVST+ configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# bridge-priority 4096
device(config-rpvst)# forward-delay 20
device(config-rpvst)# hello-time 22
device(config-rpvst)# max-age 8
device(config-rpvst)# transmit-holdcount 9
device(config-rpvst)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

For more information about configuring parameters, see the section STP parameter configuration.

Enabling and configuring R-PVST+ on an interface

Follow these steps to enable and configure R-PVST+ on an interface.

The ports and parameters can be configured individually on a system by:

1. Entering the commands in steps 1-3
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable R-PVST+.

```
device(config)# protocol spanning-tree rpvst
```

3. Enter interface configuration mode.

```
device(config-rpvst)# interface ethernet 0/3
```

4. Enable the spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

5. Configure the interface link type.

```
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
```

6. Specify the port priority to influence the selection of root or designated ports.

```
device(conf-if-eth-0/3)# spanning-tree priority 64
```

   The range of priority values is from 0 through 240 in multiples of 16. The default value is 128.

7. Configure the path cost for spanning tree calculations on the interface.

```
device(conf-if-eth-0/3)# spanning-tree cost 200000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

8. Configure the path cost for spanning tree calculations a specific VLAN.

```
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

9. Enable automatic edge detection on the interface.

```
device(conf-if-eth-0/3)# spanning-tree autoedge
```

   You use this command to automatically identify the edge port. A port becomes an edge port if it receives no BPDUs. By default, automatic edge detection is disabled.

10. Enable root guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree guard root
```

    Root guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.

11. Enable the spanning tree on the edge port.

```
device(conf-if-eth-0/3)# spanning-tree edgeport
```

    If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

12. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree edgeport bpdu-guard
```

    BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

13. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

14. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: Rapid PVST Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 ------------------------------------------------------------
 Eth 0/3      DES   FWD  200000     128   P2P            No
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 38 ≥ 25 ≥ 18.

15. Save the configuration.

```
device# copy running-config startup-config
```

R-PVST+ on an interface configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
device(conf-if-eth-0/3)# spanning-tree priority 64
device(conf-if-eth-0/3)# spanning-tree cost 200000
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
device(conf-if-eth-0/3)# spanning-tree autoedge
device(conf-if-eth-0/3)# spanning-tree guard root
device(conf-if-eth-0/3)# spanning-tree edgeport
device(conf-if-eth-0/3)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-0/3)# exit
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring R-PVST+ on a system

Follow the steps to configure R-PVST+ on a system.

The ports and parameters can be configured individually by:

1. Entering the commands in steps 1 and 2
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable R-PVST+.

```
device(config)# protocol spanning-tree rpvst
```

   You can shut down R-PVST+ by entering the **shutdown** command when in `rpvst` configuration mode.

3. Configure the bridge priority for the common instance.

```
device(config-rpvst)# bridge-priority 4096
```

   Valid values range from 0 through 61440 in increments of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the forward delay parameter.

```
device(config-rpvst)# forward-delay  20
```

5. Configure the hello time parameter.

```
device(config-rpvst)# hello-time 8
```

6. Configure the maximum age parameter.

```
device(config-rpvst)# max-age 22
```

7. Specify the transmit hold count.

```
device(config-rpvst)# transmit-holdcount 5
```

   This command configures the maximum number of BPDUs transmitted per second. The range of values is 1 through 10.

8. Configure VLANs.
   a) Configure VLAN 100 with a priority of 0.

```
device(config-rpvst)# vlan 100 priority 0
```

      Valid priority values range from 0 through 61440 in multiples of 4096.

   b) Configure VLAN 201 with a priority of 12288.

```
device(config-rpvst)# vlan 201 priority 12288
```

   c) Configure VLAN 301 with a priority of 20480.

```
device(config-rpvst)# vlan 301 priority 20480
```

9. Set the switching characteristics for interface 0/3.
   a) Enter interface configuration mode.

```
device(config-rpvst)# interface ethernet 0/3
```

   b) Set the switching characteristics of the interface.

```
device(conf-if-eth-0/3)# switchport
```

   c) Set the interface mode to trunk.

```
device(conf-if-eth-0/3)# switchport mode trunk
```

   d) Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
```

   e) Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
```

   f) Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
```

   g) Enable spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

   h) Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

10. Set the switching characteristics for interface 0/4.
    a) Enter interface configuration mode.

```
device(config-rpvst)# interface ethernet 0/4
```

    b) Set the switching characteristics of the interface.

```
device(conf-if-eth-0/4)# switchport
```

    c) Set the interface mode to trunk.

```
device(conf-if-eth-0/4)# switchport mode trunk
```

    d) Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
```

    e) Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
```

    f) Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
```

    g) Enable spanning tree on the interface.

```
device(conf-if-eth-0/4)# no spanning-tree shutdown
```

    h) Return to privileged EXEC mode.

```
device(conf-if-eth-0/4)# exit
```

11. To interoperate with switches other than VDX switches in R-PVST+ mode, you must configure the interface that is connected to that switch.
    a) Enter interface configuration mode for the port that interoperates with a VDX switch.

```
device(config)# interface ethernet 0/12
```

    b) Specify the MAC address for the device.

```
device(conf-if-eth-0/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
```

    c) Enable spanning tree on the interface.

```
device(conf-if-eth-0/12)# no spanning-tree shutdown
```

    d) Return to privileged EXEC mode.

```
device(conf-if-eth-0/12)# end
```

12. Verify the configuration.

```
device# show spanning-tree

VLAN 1

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 0001.01e0.5200.0180 (self)
 Bridge Id: 0001.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 100

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 0064.01e0.5200.0180 (self)
 Bridge Id: 0064.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
```

```
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 201

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 30c9.01e0.5200.0180 (self)
 Bridge Id: 30c9.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 301

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 512d.01e0.5200.0180 (self)
 Bridge Id: 512d.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
```

```
Number of topology change(s): 0

Bpdu-guard errdisable timeout: disabled
Bpdu-guard errdisable timeout interval: 300 sec

Port Et 0/3 enabled
   Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
   Designated Path Cost: 0
   Configured Path Cost: 20000000
   Designated Port Id: 0; Port Priority: 128
   Designated Bridge: 0000.0000.0000.0000
   Number of forward-transitions: 0
   Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
   Portfast: off
   Configured Root guard: off; Operational Root guard: off
   Bpdu-guard: off
   Link-type: point-to-point
   Received BPDUs: 0; Sent BPDUs: 0

Port Et 0/4 enabled
   Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
   Designated Path Cost: 0
   Configured Path Cost: 20000000
   Designated Port Id: 0; Port Priority: 128
   Designated Bridge: 0000.0000.0000.0000
   Number of forward-transitions: 0
   Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
   Portfast: off
   Configured Root guard: off; Operational Root guard: off
   Bpdu-guard: off
   Link-type: point-to-point
   Received BPDUs: 0; Sent BPDUs: 0
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

13. Save the configuration.

```
device# copy running-config startup-config
```

Enable R-PVST+ on a system configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# bridge-priority 4096
device(config-rpvst)# forward-delay  20
device(config-rpvst)# hello-time 8
device(config-rpvst)# max-age 22
device(config-rpvst)# transmit-holdcount 5
device(config-rpvst)# vlan 100 priority 0
device(config-rpvst)# vlan 201 priority 12288
device(config-rpvst)# vlan 301 priority 20480
device(config-rpvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# switchport
device(conf-if-eth-0/3)# switchport mode trunk
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# exit
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# switchport
device(conf-if-eth-0/4)# switchport mode trunk
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/4)# no spanning-tree shutdown
device(conf-if-eth-0/4)# end
device# show spanning-tree
device# copy running-config startup-config
```

Clearing spanning tree counters

Follow these steps to clear spanning tree counters on all interfaces or on the specified interface.

1. Clear spanning tree counters on all interfaces.

   ```
   device# clear spanning-tree counter
   ```

2. Clear spanning tree counters on a specified Ethernet interface.

   ```
   device# clear spanning-tree counter interface ethernet 0/3
   ```

3. Clear spanning tree counters on a specified port channel interface.

   ```
   device# clear spanning-tree counter interface port-channel 12
   ```

   Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

Follow these steps to restart the protocol migration process.

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

   ```
   device# clear spanning-tree detected-protocols
   ```

2. Restart the spanning tree migration process on a specific Ethernet interface.

   ```
   device# clear spanning-tree detected-protocols interface ethernet 0/3
   ```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

   Port channel interface numbers range from 1 through 64.

Shutting down PVST+ or R-PVST+

Follow these steps to o shut down PVST+, or R-PVST+ either globally, on a specific interface, or a specific VLAN.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down PVST+ or R-PVST+.
   - Shut down PVST+ or R-PVST+ globally and return to privileged EXEC mode.

```
device(config)# protocol spanning-tree pvst
device(config-pvst)# shutdown
device(config-pvst)# end
```

   - Shut down PVST+ or R-PVST+ on a specific interface and return to privileged EXEC mode.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree shutdown
device(conf-if-eth-0/2)# end
```

   - Shut down PVST+ or R-PVST+ on a specific VLAN. and return to privileged EXEC mode.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
```

3. Verify the configuration.

```
device# show spanning-tree
device#
```

4. Save the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

Shut down PVST+ or R-PVST+ configuration example

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
device# show spanning-tree
device# copy running-config startup-config
```

NOTE

Shutting down PVST+ on a VLAN is used in this example.

802.1s Multiple Spanning Tree Protocol

- MSTP overview 155
- MSTP global level parameters 157
- MSTP interface level parameters 158
- Configuring MSTP 159

MSTP overview

IEEE 802.1s Multiple STP (MSTP) helps create multiple loop-free active topologies on a single physical topology.

MSTP uses RSTP to group VLANs into separate spanning-tree instance. Each instance has its own spanning-tree topology independent of other spanning tree instances, which allows multiple forwarding paths, permits load balancing, and facilitates the movement of data traffic. A failure in one instance does not affect other instances. By enabling the MSTP, you are able to more effectively utilize the physical resources present in the network and achieve better load balancing of VLAN traffic.

The MSTP evolved as a compromise between the two extremes of the RSTP and R-PVST+, it was standardized as IEEE 802.1s and later incorporated into the IEEE 802.1Q-2003 standard. The MSTP configures a meshed topology into a loop-free, tree-like topology. When the link on a bridge port goes up, an MSTP calculation occurs on that port. The result of the calculation is the transition of the port into either a forwarding or blocking state. The result depends on the position of the port in the network and the MSTP parameters. All the data frames are forwarded over the spanning tree topology calculated by the protocol.

NOTE

Multiple switches must be configured consistently with the same MSTP configuration to participate in multiple spanning tree instances. A group of interconnected switches that have the same MSTP configuration is called an MSTP region.
MSTP is backward compatible with the STP and the RSTP.

Common Spanning Tree (CST)

The single Spanning Tree instance used by the Extreme device, and other vendor devices to interoperate with MSTP bridges. This spanning tree instance stretches across the entire network domain (including PVST, PVST+ and MSTP regions). It is associated with VLAN 1 on the Extreme device.

Internal Spanning Tree (IST)

An MSTP bridge must handle at least these two instances: one IST and one or more MSTIs (Multiple Spanning Tree Instances). Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance known as IST, which extends CST inside the MST region. IST always exists if the device runs MSTP. Besides IST, this implementation supports up to 31 MSTIs.

Common Internal Spanning Tree (CIST)

The single spanning tree calculated by STP (including PVST+) and RSTP (including R-PVST+) and the logical continuation of that connectivity through MSTP bridges and regions, calculated by MSTP to ensure that all LANs in the bridged LAN are simply and fully connected

Multiple Spanning Tree Instance (MSTI)

One of a number of spanning trees calculated by the MSTP within an MST Region, to provide a simply and fully connected active topology for frames classified as belonging to a VLAN that is mapped to the MSTI by the MST configuration table used by the MST bridges of that MST region.

The Extreme implementation supports up to 32 spanning tree instances in an MSTP enabled bridge that can support up to 32 different Layer 2 topologies. The spanning tree algorithm used by the MSTP is the RSTP, which provides quick convergence.

By default all configured VLANs including the default VLAN are assigned to and derive port states from CIST until explicitly assigned to MSTIs.

MST regions

MST regions are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels. Also, one or more VLANs can be mapped to one MST instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances

MSTP regions

MSTP introduces a hierarchical way of managing device domains using regions. Devices that share common MSTP configuration attributes belong to a region. The MSTP configuration determines the MSTP region where each device resides. The common MSTP configuration attributes are as follows:

- Alphanumeric configuration name (32 bytes)
- Configuration revision number (2 bytes)
- 4096-element table that maps each of the VLANs to an MSTP instance

Region boundaries are determined by the above attributes. An MSTI is an RSTP instance that operates inside an MSTP region and determines the active topology for the set of VLANs mapping to that instance. Every region has a CIST that forms a single spanning tree instance which includes all the devices in the region. The difference between the CIST instance and the MSTP instance is that the CIST instance operates across the MSTP region and forms a loop-free topology across regions, while the MSTP instance operates only within a region. The CIST instance can operate using the RSTP only if all the devices across the regions support the RSTP. However, if any of the devices operate using the STP, the CIST instance reverts to the STP.

Each region is viewed logically as a single STP or a single RSTP bridge to other regions.

NOTE
Extreme supports 32 MSTP instances and one MSTP region.

For more information about spanning trees, see the introductory sections in the Spanning Tree Protocol chapter.

MSTP guidelines and restrictions

Follow these restrictions and guidelines when configuring the MSTP:

- Create VLANs before mapping them to the MSTP instances.
- The Extreme implementation of the MSTP supports up to 32 MSTP instances and one MSTP region.
- The MSTP **force-version** option is not supported.
- You must create VLANs before mapping them to the MSTP instances.

- For two or more switches to be in the same the MSTP region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same region name.
- MSTP is backward compatible with the STP and the RSTP.
- Only one MSTP region can be configured on a bridge.
- A maximum of 4090 VLANs can be configured across the 32 MSTP instances.
- MSTP and topology groups cannot be configured together.

Default MSTP configuration

As well as the defaults listed in the section Understanding the default STP configuration on page 103 there are defaults that apply only to MSTP configurations.

Parameter	Default setting
Cisco interoperability	Disabled
Device priority (when mapping a VLAN to an MSTP instance)	32768
Maximum hops	20 hops
Revision number	0

Interoperability with PVST+ and R-PVST+

Since Extreme or other vendor devices enabled with PVST+ and R-PVST+ send IEEE STP BPDUs in addition to the PVST and R-PVST BPDUs, the VLAN 1 spanning tree joins the Common Spanning Tree (CST) of the network and thus interoperates with MSTP. The IEEE compliant devices treat the BPDUs addressed to the Extreme proprietary multicast MAC address as an unknown multicast address and flood them over the active topology for the particular VLAN.

MSTP global level parameters

To configure a switch for MSTP, first you set the region name and the revision on each switch that is being configured for MSTP. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments.

Each of the steps used to configure and operate MSTP are described in the following:

NOTE
The MSTP Region and Revision global parameters are enabled for interface level parameters as described below.

- Set the MSTP region name — Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions. The default MSTP name is "NULL".
- Set the MSTP revision number — Each switch that is running MSTP is configured with a revision number. It applies to the switch, which can have many different VLANs that can belong to many different MSTP regions.
- Enabling and disabling Cisco interoperability — While in MSTP mode, use the **cisco-interoperability** command to enable or disable the ability to interoperate with certain legacy Cisco switches. If Cisco interoperability is required on any switch in the network, then all switches in the network must be compatible, and therefore enabled by means of this command. By default the Cisco interoperability is disabled.

- The parameters you would normally set when you configure STP are applicable to MSTP. Before you configure MSTP parameters see the sections explaining bridge parameters, the error disable timeout parameter and the port-channel path cost parameter in the STP section of this guide.

MSTP interface level parameters

Edge port and automatic edge detection

Configuring the edge port feature makes a port transition directly from initialization to the forwarding state, skipping the listening and learning states.

From an interface, you can configure a device to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

Follow these guidelines to configure a port as an edge port:

- When edge port is enabled, the port still participates in a spanning tree.
- A port can become an edge port if no BPDU is received.
- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

  NOTE
  If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

BPDU guard

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow.

In a valid configuration, edge port-configured interfaces do not receive BPDUs. If an edge port-configured interface receives a BPDU, an invalid configuration exists, such as the connection of an unauthorized device. The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service.

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Extreme device port to which the end station is connected. The STP BPDU guard shuts down the port and puts it into an "error disabled" state. This disables the connected device's ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a message is displayed to warn the network administrator of an invalid configuration.

The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service with the **no shutdown** command if error disable recovery is not enabled by enabling the **errdisable-timeout** command. The interface can also be automatically configured to be enabled after a timeout. However, if the offending BPDUs are still being received, the port is disabled again.

Expected behavior in an interface context

When BPDU Guard is enabled on an interface, the device is expected to put the interface in Error Disabled state when BPDU is received on the port when edge-port and BPDU guard is enabled on the switch interface. When the port ceases to receive the BPDUs, it does not automatically switch to edge port mode, you must configure **error disable timeout** or **no shutdown** on the port to move the port back into edge port mode.

Restricted role

Configuring restricted role on a port causes the port not to be selected as root port for the CIST or any MSTI, even if it has the best spanning tree priority vector.

Restricted role ports are selected as an alternate port after the root port has been selected. It is configured by a network administrator to prevent bridges external to a core region of the network influencing the spanning tree active topology, possibly because those bridges are not under the full control of the administrator. It will protect the root bridge from malicious attack or even unintentional misconfigurations where a bridge device which is not intended to be root bridge, becomes root bridge causing severe bottlenecks in data path. These types of mistakes or attacks can be avoided by configuring 'restricted-role' feature on ports of the root bridge . This feature is similar to the "root-guard" feature which is proprietary implementation of Cisco for STP and RSTP but had been adapted in the 802.1Q standard as "restricted-role". The "restricted-role" feature if configured on an incorrect port can cause lack of spanning tree connectivity.

Expected behavior in an interface context

When this feature is enabled on an interface the device is expected to prevent a port configured with restricted-role feature from assuming the role of a Root port. Such a port is expected to assume the role of an Alternate port instead, once Root port is selected.

Restricted TCN

TCN BPDUs are used to inform other switches of port changes.

Configuring "restricted TCN" on a port causes the port not to propagate received topology change notifications and topology changes originated from a bridge external to the core network to other ports. It is configured by a network administrator to prevent bridges external to a core region of the network from causing MAC address flushing in that region, possibly because those bridges are not under the full control of the administrator for the attached LANs. If configured on an incorrect port it can cause temporary loss of connectivity after changes in a spanning trees active topology as a result of persistent incorrectly learned station location information.

Expected behavior in an interface context

When this feature is enabled on an interface, the device is expected to prevent propagation of topology change notifications from a port configured with the Restricted TCN feature to other ports. In this manner, the device prevents TCN propagation from causing MAC flushes in the entire core network.

Configuring MSTP

Enabling and configuring MSTP globally

Follow this procedure to configure the Multiple Spanning Tree Protocol.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable MSTP.

```
device(config)# protocol spanning-tree mstp
```

This command creates a context for MSTP. MSTP is automatically enabled. All MSTP specific CLI commands can be issued only from this context. Entering **no protocol spanning-tree mstp** deletes the context and all the configurations defined within the context.

3. Specify the region name.

```
device(config-mstp)# region kerry
```

4. Specify the revision number.

```
device(config-mstp)# revision 1
```

5. Configure an optional description of the MSTP instance.

```
device(config-mstp)# description kerry switches
```

6. Specify the maximum hops for a BPDU to prevent the messages from looping indefinitely on the interface.

```
device(config-mstp)# max-hops 25
```

Setting this parameter prevents messages from looping indefinitely on the interface. The range is 1 through 40 hops while the default is 20 .

7. Map VLANs to MSTP instances and set the instance priority.
   a) Map VLANs 7 and 8 to instance 1.

```
device(config-mstp)# instance 1 vlan 7,8
```

   b) Map VLANs 21, 22, and 23 to instance 2.

```
device(config-mstp)# instance 2 vlan 21-23
```

   c) Set the priority of instance 1.

```
device(config-mstp)# instance 1 priority 4096
```

This command can be used only after the VLAN is created. VLAN instance mapping is removed from the configuration if the underlying VLANs are deleted.

8. Configure a bridge priority for the CIST bridge.

```
device(config-mstp)# bridge-priority 4096
```

The range is 0 through 61440 in increments of 4096. The default is 32768.

9. Set the error disable parameters.
   a) Enable the timer to bring the port out of error disable state.

      ```
      device(config-mstp)# error-disable-timeout enable
      ```

   b) Specify the time in seconds it takes for an interface to time out..

      ```
      device(config-mstp)# error-disable-timeout interval 60
      ```

      The range is fro 10 to 1000000 seconds with a default of 300 seconds.
10. Configure forward delay.
    a) Specify the bridge forward delay.

       ```
       device(config-mstp)# forward-delay 15
       ```

    This command allows you to specify how long an interface remains in the listening and learning states before it begins forwarding. This command affects all MSTP instances. The range of values is from 4 to 30 seconds with a default of 15 seconds.
11. Configure hello time.

    ```
    device(config-mstp)# hello-time 2
    ```

    The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The range is from 1 through 10 seconds with a default of 2 seconds.
12. Configure the maximum age.

    ```
    device(config-mstp)# max-age 20
    ```

    You must set the **max-age** so that it is greater than the **hello-time**. The range is 6 through 40 seconds with a default of 20 seconds.
13. Specify the port-channel path cost.

    ```
    device(config-mstp)# port-channel path-cost custom
    ```

    This command allows you to control the path cost of a port channel according to bandwidth.
14. Specify the transmit hold count.

    ```
    device(config-mstp)# transmit-holdcount 5
    ```

    The transmit hold count is used to limit the maximum number of MSTP BPDUs that the bridge can transmit on a port before pausing for 1 second. The range is from 1 to 10 seconds with a default of 6 seconds.
15. Configure Cisco interoperability.

    ```
    device(config-mstp)# cisco-interoperability enable
    ```

    This command enables the ability to interoperate with certain legacy Cisco switches. The default is Cisco interoperability is disabled.
16. Return to privileged exec mode.

    ```
    device(config-mstp)# end
    ```

17. Verify the configuration. The following is an example configuration.

```
device# show spanning-tree mst-config

Spanning-tree Mode: Multiple Spanning Tree Protocol

 CIST Root Id: 8000.001b.ed9f.1700
 CIST Bridge Id: 8000.768e.f80a.6800
 CIST Reg Root Id: 8000.001b.ed9f.1700

 CIST Root Path Cost: 0; CIST Root Port: Eth 0/2
 CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 19
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20;
 Tx-HoldCount: 6
 Number of topology change(s): 139; Last change occurred 00:03:36 ago on Eth 0/2

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec
 Migrate Time: 3 sec

  Name           : kerry
  Revision Level : 1
  Digest         : 0x9357EBB7A8D74DD5FEF4F2BAB50531AA

  Instance        VLAN
  --------        ----
   0:             1
   1:             7,8
   2:             21-23
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

18. Save the configuration.

```
device# copy running-config startup-config
```

MSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# region kerry
device(config-mstp)# revision 1
device(config-mstp)# description kerry switches
device(config-mstp)# max-hops 20
device(config-mstp)# instance 1 vlan 7,8
device(config-mstp)# instance 2 vlan 21-23
device(config-mstp)# instance 1 priority 4096
device(config-mstp)# bridge-priority 4096
device(config-mstp)# error-disable-timeout enable
device(config-mstp)# error-disable-timeout interval 60
device(config-mstp)# forward-delay 16
device(config-mstp)# hello-time 5
device(config-mstp)# max-age 16
device(config-mstp)# port-channel path-cost custon
device(config-mstp)# transmit-holdcount 5
device(config-mstp)# cisco-interoperability enable
device(config-mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Enabling and configuring MSTP on an interface

Follow these steps to configure and enable MSTP on an Ethernet interface.

The parameters can be configured individually on an interface by:

1. Entering the commands in Steps 1 through Step 3 for the target interface
2. Running the relevant parameter command
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter configuration mode.

```
device# configure terminal
```

2. Enable MSTP.

```
device(config)# protocol spanning-tree mstp
```

3. Enter interface configuration mode.

```
device(config-mstp)# interface ethernet 0/5
```

4. Enable the interface.

```
device(conf-if-eth-0/5)# no shutdown
```

5. Configure the restricted role feature for the port.

```
device(conf-if-eth-0/5)# spanning-tree restricted-role
```

   This command keeps a port from becoming a root.

6. Restrict topology change notifications (TCN) BPDUs for an MSTP instance.

```
device(conf-if-eth-0/5)# spanning-tree instance 5 restricted-tcn
```

   This prevents the port from propagating received TCNs and topology changes originating from a bridge, external to the core network, to other ports.

7. Enable auto detection of an MSTP edge port.

```
device(conf-if-eth-0/5)# spanning-tree autoedge
```

   Enabling this feature allows the system to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

8.

```
device(conf-if-eth-0/5)# spanning-tree edgeport
```

   Enabling edge port allows the port to quickly transition to the forwarding state. By default, automatic edge detection is disabled.

9. Enable BPDU guard on the port

```
device(conf-if-eth-0/5)# spanning-tree edgeport bpdu-guard
```

   BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

10. Set the path cost of a port.

```
device(conf-if-eth-0/5)# spanning-tree cost 200000
```

    The path cost range is from 1 to 200000000. Leaving the default adjusts path cost relative to changes in the bandwidth. A lower path cost indicates greater likelihood of becoming root port.

11. Configure the link type.

```
device(conf-if-eth-0/5)# spanning-tree link-type point-to-point
```

    The options are point-to-point or shared.

12. Enable port priority.

```
device(conf-if-eth-0/5)# spanning-tree priority 128
```

    The range is from 0 to 240 in increments of 16 with a default of 32. A lower priority indicates greater likelihood of becoming root port.

13. Return to privileged exec mode.

```
device(conf-if-eth-0/5)# end
```

14. Verify the configuration.

```
device# show spanning-tree interface ethernet 0/5

Spanning-tree Mode: Multiple Spanning Tree Protocol

 Root Id: 8000.001b.ed9f.1700
 Bridge Id: 8000.01e0.5200.011d

 Port Eth 0/5 enabled
    Ifindex: 411271175; Id: 8002; Role: Designated; State: Forwarding
    Designated External Path Cost: 0; Internal Path Cost: 20000000
    Configured Path Cost: 200000
    Designated Port Id: 8002; Port Priority: 128
    Designated Bridge: 8000.01e0.5200.011d
    Number of forward-transitions: 1
    Version: Multiple Spanning Tree Protocol - Received MSTP - Sent MSTP
    Edgeport: yes; AutoEdge: yes; AdminEdge: no; EdgeDelay: 3 sec
    Restricted-role is enabled
    Restricted-tcn is enabled
    Boundary: no
    Bpdu-guard: on
    Link-type: point-to-point
    Received BPDUs: 86; Sent BPDUs: 1654
```

15. Save the configuration.

```
device# copy running-config startup-config
```

Enable MSTP on an interface configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# interface ethernet 0/5
device(conf-if-eth-0/5)# no shutdown
device(conf-if-eth-0/5)# spanning-tree restricted-role
device(conf-if-eth-0/5)# spanning-tree instance 5 restricted-tcn
device(conf-if-eth-0/5)# spanning-tree autoedge
device(conf-if-eth-0/5)# spanning-tree edgeport
device(conf-if-eth-0/5)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-0/5)# spanning-tree cost 200000
device(conf-if-eth-0/5)# spanning-tree link-type point-to-point
device(conf-if-eth-0/5)# spanning-tree priority 128
device(conf-if-eth-0/5)# end
device# show spanning-tree interface ethernet 0/5
device# copy running-config startup-config
```

Enabling MSTP on a VLAN

1. Enter configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the protocol command to enable MSTP configuration.

   ```
   device(config)# protocol spanning-tree mstp
   ```

3. Map a VLAN to an MSTP instance.

   ```
   device(config-mstp)# instance 5 vlan 300
   ```

4. Return to privileged EXEC mode.

   ```
   device(config-mstp)# end
   ```

5. Verify the configuration.

```
device# show spanning-tree mst

 Spanning-tree Mode: Multiple Spanning Tree Protocol

 CIST Root Id: 8000.609c.9f5d.4800 (self)
 CIST Bridge Id: 8000.609c.9f5d.4800
 CIST Reg Root Id: 8000.609c.9f5d.4800 (self)

 CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20;
 Tx-HoldCount: 6
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec
 Migrate Time: 3 sec

  Name           : NULL
  Revision Level : 0
  Digest         : 0xD5FF4C3F6C18E2F27AF3A8300297ABAA

  Instance        VLAN
  --------        ----
   0:             1
   5:             100
```

   Observe that the settings comply with the formula set out in the STP parameters section, as:

   (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

   or in this case: 28 ≥ 20 ≥ 6.

6. Save the configuration.

```
device# copy running-config startup-config
```

Enable spanning tree on a VLAN configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# instance 5 vlan 300
device(config-mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Configuring basic MSTP parameters

Follow these steps to configure basic MSTP parameters.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable MSTP.

```
device(config)# protocol spanning-tree mstp
```

3. Specify the region name.

```
device(config-mstp)# region connemara
```

4. Specify the revision number.

```
device(config-mstp)# revision 1
```

5. Map MSTP instances to VLANs.
   a) Map instance 1 to VLANs 2 and 3.

```
device(config-mstp)# instance 1 vlan 2,3
```

   b) Map instance 2 to VLANs 4, 5, and 6.

```
device(config-mstp)# instance 2 vlan 4-6
```

6. Set a priority for an instance.

```
device(conf-Mstp)# instance 1 priority 28672
```

   The priority ranges from 0 through 61440 and the value must be in multiples of 4096.

7. Specify the maximum hops for a BPDU.

```
device(conf-Mstp)# max-hops 25
```

   This prevents the messages from looping indefinitely on an interface

8. Return to privileged EXEC mode.

```
device(conf-Mstp)# end
```

9. Verify the configuration.

```
device# show spanning-tree mst

 Spanning-tree Mode: Multiple Spanning Tree Protocol

 CIST Root Id: 8000.609c.9f5d.4800 (self)
 CIST Bridge Id: 8000.609c.9f5d.4800
 CIST Reg Root Id: 8000.609c.9f5d.4800 (self)

 CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 25;
 Tx-HoldCount: 6
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec
 Migrate Time: 3 sec

  Name           : connemara
  Revision Level : 1
  Digest         : 0xD5FF4C3F6C18E2F27AF3A8300297ABAA

  Instance        VLAN
  --------        ----
   0:             1,7,8,9
   1:             2,3
   2:             4-6
```

NOTE

Observe that the settings comply with the formula set out in the STP parameters section, as:
(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))
or in this case: 28 ≥ 20 ≥ 6.

```
device# show running-config | begin spanning-tree
protocol spanning-tree mstp
 instance 1 vlan 2,3
 instance 1 priority 28672
 instance 2 vlan 4-6
 region connemars
 revision 1
 max-hops 25
 !
...
```

10. Save the configuration

```
device# copy running-config startup-config
```

Basic MSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# region connemara
device(config-mstp)# revision 1
device(config-mstp)# instance 1 vlan 2,3
device(config-mstp)# instance 2 vlan 4-6
device(conf-Mstp)# instance 1 priority 28582
device(conf-Mstp)# max-hops 25
device(conf-Mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Clearing spanning tree counters

Follow these steps to clear spanning tree counters on all interfaces or on the specified interface.

1. Clear spanning tree counters on all interfaces.

```
device# clear spanning-tree counter
```

2. Clear spanning tree counters on a specified Ethernet interface.

```
device# clear spanning-tree counter interface ethernet 0/3
```

3. Clear spanning tree counters on a specified port channel interface.

```
device# clear spanning-tree counter interface port-channel 12
```

   Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

Follow these steps to restart the protocol migration process.

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

```
device# clear spanning-tree detected-protocols
```

2. Restart the spanning tree migration process on a specific Ethernet interface.

```
device# clear spanning-tree detected-protocols interface ethernet 0/3
```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

   Port channel interface numbers range from 1 through 64.

Shutting down MSTP

Follow these steps to o shut down MSTP either globally, on a specific interface, or a specific VLAN.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down MSTP.
   - Shut down MSTP globally and return to privileged EXEC mode.

     ```
     device(config)# protocol spanning-tree mstp
     device(config-mstp)# shutdown
     device(config-mstp)# end
     ```

   - Shut down MSTP on a specific interface and return to privileged EXEC mode.

     ```
     device(config)# interface ethernet 0/2
     device(conf-if-eth-0/2)# spanning-tree shutdown
     device(conf-if-eth-0/2)# end
     ```

   - Shut down MSTP on a specific VLAN and return to privileged EXEC mode.

     ```
     device(config)# vlan 10
     device(config-vlan-10)# spanning-tree shutdown
     device(config-vlan-10)# end
     ```

3. Verify the configuration.

   ```
   device# show spanning-tree
   device#
   ```

4. Save the running configuration to the startup configuration.

   ```
   device# copy running-config startup-config
   ```

Shut down MSTP configuration example

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-stp)# end
device# show spanning-tree
device# copy running-config startup-config
```

NOTE

Shutting down MSTP on a VLAN is used in this example.

Link-Fault Signaling (LFS)

- LFS overview 171
- Configuring LFS globally 171
- Configuring LFS at interface level 172
- LFS show and clear commands 172

LFS overview

Link-fault signaling (LFS) is a physical layer protocol that shuts down an interface if there is a link fault.

LFS is always enabled for egress traffic and cannot be disabled.

For ingress traffic, LFS is enabled by default at device level. You have the following options to modify this default setting:

- You can disable and enable LFS at device level.
- You can disable and enable LFS for each ethernet interface.

NOTE
LFS settings at interface level override the global setting.

When LFS is enabled, if there is an ingress link fault, the affected interface is brought down. When LFS is disabled, if the PHY-MAC link is up, the interface stays up—even if there is an ingress link fault.

LFS configuration guidelines.

Follow these guidelines when implementing link-fault signaling (LFS):

- LFS is supported both in default system mode and network packet broker (NPB) mode.
- The LFS state of a port-channel is governed by the LFS state of its aggregated physical interfaces.

Configuring LFS globally

Use this task to change the global link-fault signaling (LFS) setting.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the relevant **link-fault-signaling rx** command.
   - To change the default global LFS ingress setting from on to off, enter **link-fault-signaling rx off**.

```
device(config)# link-fault-signaling rx off
```

   - To restore the global LFS ingress setting to on, enter **no link-fault-signaling**.

```
device(config)# no link-fault-signaling
```

Configuring LFS at interface level

Use this task to enable or disable link-fault signaling (LFS) on an ethernet interface.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to access interface configuration mode.

```
device(config)# interface ethernet 0/5
```

3. Enter the relevant **link-fault-signaling rx** command.
   - To set interface LFS to off—overriding the global setting—enter **link-fault-signaling rx off**.

```
device(conf-if-eth-0/5)# link-fault-signaling rx off
```

   - To set interface LFS to configured on—overriding the global setting—enter **link-fault-signaling rx on**.

```
device(conf-if-eth-0/5)# link-fault-signaling rx on
```

   - To set interface LFS to default on—but able to be overridden by a global **link-fault-signaling rx off**—enter **no link-fault-signaling**.

```
device(conf-if-eth-0/5)# no link-fault-signaling
```

LFS show and clear commands

There is a full range of LFS show and clear commands, listed here with descriptions.

TABLE 28 LFS show commands in the *Command Reference*

Command	Description
show link-fault-signaling	Displays the global and interface link-fault signaling (LFS) statuses.
show link-fault-signaling counters	Displays LFS counter information.

TABLE 29 LFS clear commands in the *Command Reference*

Command	Description
clear counters link-fault-signaling	Clears LFS counter information for all interfaces or for a specified interface.