

Extreme Network OS Layer 2 Switching Configuration Guide

Supporting Network OS 7.3.0

© 2018, Extreme Networks, Inc. All Rights Reserved.

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 are the property of their respective owners. For additional information on Extreme Networks Trademarks please see www.extremenetworks.com/company/legal/trademarks. Specifications and product availability are subject to change without notice.

Contents

Preface	7
Document conventions.....	7
Notes, cautions, and warnings.....	7
Text formatting conventions.....	7
Command syntax conventions.....	8
Extreme resources.....	8
Document feedback.....	8
Contacting Extreme Technical Support.....	9
About this document	11
Supported hardware and software.....	11
Using the Network OS CLI	11
What's new in this document.....	11
FCoE	13
FCoE overview.....	13
FCoE terminology.....	13
End-to-end FCoE.....	14
FCoE and Layer 2 Ethernet.....	16
FCoE Initialization Protocol	21
FCoE queuing.....	24
FCoE logical SAN overview.....	24
FCoE logical SAN use cases.....	25
FCoE logical SAN behavior and provisioning model.....	28
FCoE logical SAN limitations.....	32
FCoE logical SAN upgrade/downgrade considerations.....	32
FCoE logical SAN scalability.....	33
FCoE logical SAN configuration recommendations.....	34
Configuring FCoE.....	34
Configuring logical FCoE ports.....	34
Configuring fabric maps.....	35
Configuring FCoE logical SANs.....	35
Managing duplicate WWNs.....	47
802.1Q VLANs	49
802.1Q VLAN overview.....	49
Ingress VLAN filtering.....	49
VLAN configuration guidelines and restrictions.....	51
Configuring and managing 802.1Q VLANs.....	51
Understanding the default VLAN configuration.....	51
Configuring interfaces to support VLANs.....	52
Configuring protocol-based VLAN classifier rules.....	55
Displaying VLAN information.....	57
Configuring the MAC address table and conversational MAC learning.....	57
Private VLANs.....	60
PVLAN configuration guidelines and restrictions.....	60
Associating the primary and secondary VLANs.....	61
Configuring an interface as a PVLAN promiscuous port.....	61

Configuring an interface as a PVLAN host port.....	62
Configuring an interface as a PVLAN trunk port.....	62
Displaying PVLAN information.....	63
Protected port for uplinks	63
VLAN Layer 2 forwarding.....	64
Protected port behavior.....	65
Limitations and considerations.....	65
Configuring protected port for uplinks.....	66
VXLAN Overlay Gateways for NSX Controller Deployments.....	67
Introduction to VXLAN overlay gateways with NSX Controller.....	67
VXLAN NSX replicator load balancing.....	68
Configuring a VXLAN overlay gateway for NSX Controller deployments.....	68
High-level communication in a VXLAN environment with an NSX Controller.....	68
Coordination of activities in NSX Controller deployments.....	69
Configuring VRRP-E for NSX Controller deployments.....	70
Configuring a loopback interface VTEP for NSX Controller deployments.....	71
VXLAN gateway and NSX Controller deployments.....	72
Configuring VXLAN NSX replicator load balancing.....	76
Additional commands for VXLAN configuration.....	77
Distributed VXLAN Gateways.....	79
Distributed VXLAN gateways overview.....	79
Distributed VXLAN gateways supported topologies.....	79
Distributed VXLAN gateways unsupported topologies.....	81
Distributed VXLAN gateways RBridge scalability.....	83
Distributed VXLAN gateways upgrade and downgrade considerations.....	83
Distributed VXLAN gateways limitations.....	84
Configuring a distributed VXLAN gateway.....	84
Troubleshooting and managing distributed VXLAN gateways.....	85
Troubleshooting.....	85
STP-Type Protocols.....	87
STP overview.....	87
STP configuration guidelines and restrictions.....	87
RSTP.....	88
MSTP.....	89
PVST+ and Rapid PVST+	90
Spanning Tree Protocol and VCS mode.....	90
Configuring and managing STP and STP variants.....	91
Understanding the default STP configuration.....	91
Configuring STP.....	92
Configuring RSTP	93
Configuring MSTP	94
Configuring PVST+ or R-PVST+.....	97
Enabling STP, RSTP, MSTP, PVST+ or R-PVST+.....	97
Shutting down STP, RSTP, MSTP, PVST+, or R-PVST+ globally.....	98
Specifying bridge parameters.....	98
Configuring STP timers.....	101
Specifying the port-channel path cost.....	101
Specifying the transmit hold count (RSTP, MSTP, and R-PVST+).....	102
Clearing spanning tree counters.....	102

Clearing spanning tree-detected protocols.....	103
Displaying STP, RSTP, MSTP, PVST+, or R-PVST+ information.....	103
Configuring STP, RSTP, or MSTP on DCB interface ports.....	103
Configuring DiST.....	109
Cisco Peer-Switch support.....	110
UDLD.....	113
UDLD overview.....	113
UDLD requirements.....	113
How UDLD works.....	113
Configuring UDLD.....	114
Additional UDLD-related commands.....	115
Link Aggregation	117
Link aggregation overview.....	117
Link Aggregation Control Protocol.....	118
Extreme-proprietary aggregation.....	118
LAG distribution process and conditions.....	118
Virtual LAGs	119
IP over port-channel.....	120
Ethernet Segment Identifiers (ESIs) for BGP routing.....	124
Link aggregation setup.....	124
vLAG configuration overview.....	124
Configuring load balancing on a remote RBridge.....	132
Configuring and managing Link Aggregation.....	133
AMPP.....	139
AMPP overview.....	139
AMPP over vLAG	139
AMPP and Switched Port Analyzer	140
AMPP scalability.....	141
AMPP port-profiles	141
Configuring AMPP profiles.....	143
Configuring a new port-profile.....	143
Configuring VLAN profiles.....	144
Configuring FCoE profiles.....	145
Configuring QoS profiles.....	145
Configuring security profiles.....	146
Creating a port-profile-port.....	147
Deleting a port-profile-port	147
Deleting a port-profile.....	147
Deleting a sub-profile.....	148
Creating a new port-profile domain and adding port profiles.....	148
Monitoring AMPP profiles.....	149
Link-State Tracking (LST).....	151
Link-State Tracking (LST) overview.....	151
Redundant-link topology.....	151
LST operation.....	152
General configuration guidelines for LST	153
Configuring LST for independent RBridges.....	153
Configuring LST for single-link topologies.....	154

Configuring LST for multiple-uplink topologies.....	154
Configuring LST for multiple downlink/uplink topologies	155
Configuring LST for VCS fabrics.....	156
VCS redundant-link topology.....	157
LST configuration guidelines under VCS.....	158
Configuring LST on a VCS cluster	158
Configuring LST on a VCS cluster and an independent RBridge.....	159
Disabling LST.....	160
LST show commands.....	161
Unicast Reverse Path Forwarding (uRPF).....	163
uRPF overview.....	163
Devices supported for uRPF.....	163
uRPF configuration guidelines.....	163
uRPF implementation.....	164
Configuring uRPF on a physical interface.....	164
Configuring uRPF on a port-channel interface.....	164
Configuring uRPF on a VE interface.....	165
uRPF show commands	165

Preface

- Document conventions..... 7
- Extreme resources..... 8
- Document feedback..... 8
- Contacting Extreme Technical Support..... 9

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Extreme technical documentation.

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.
<i>italic text</i>	Identifies text to enter in the GUI. Identifies emphasis. Identifies variables.
Courier font	Identifies document titles. Identifies CLI output.

Format	Description
	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.
<i>italic text</i>	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, <i>member[member...]</i> .
\	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.

Extreme resources

Visit the Extreme website to locate related documentation for your product and additional Extreme resources.

White papers, data sheets, and the most recent versions of Extreme software and hardware manuals are available at www.extremenetworks.com. Product documentation for all supported releases is available to registered users at www.extremenetworks.com/support/documentation.

Document feedback

Quality is our first concern at Extreme, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you.

You can provide feedback in two ways:

- Use our short online feedback form at <http://www.extremenetworks.com/documentation-feedback-pdf/>
- Email us at internalinfodev@extremenetworks.com

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.

Contacting Extreme Technical Support

As an Extreme customer, you can contact Extreme Technical Support using one of the following methods: 24x7 online or by telephone. OEM customers should contact their OEM/solution provider.

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

- [GTAC \(Global Technical Assistance Center\)](#) for immediate support
 - Phone: 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.
 - Email: support@extremenetworks.com. To expedite your message, enter the product name or model number in the subject line.
- [GTAC Knowledge](#) - Get on-demand and tested resolutions from the GTAC Knowledgebase, or create a help case if you need more guidance.
- [The Hub](#) - A forum for Extreme customers to connect with one another, get questions answered, share ideas and feedback, and get problems solved. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
- [Support Portal](#) - Manage cases, downloads, service contracts, product licensing, and training and certifications.

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

About this document

- Supported hardware and software..... 11
- Using the Network OS CLI 11
- What's new in this document..... 11

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 Network OS, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release of Network OS:

- ExtremeSwitching VDX 2746
- ExtremeSwitching VDX 6740
 - ExtremeSwitching VDX 6740-48
 - ExtremeSwitching VDX 6740-64
- ExtremeSwitching VDX 6740T
 - ExtremeSwitching VDX 6740T-48
 - ExtremeSwitching VDX 6740T-64
 - ExtremeSwitching VDX 6740T-1G
- ExtremeSwitching VDX 6940-36Q
- ExtremeSwitching VDX 6940-144S
- ExtremeSwitching VDX 8770
 - ExtremeSwitching VDX 8770-4
 - ExtremeSwitching VDX 8770-8

To obtain information about a Network OS version other than this release, refer to the documentation specific to that version.

Using the Network OS CLI

For complete instructions and support for using the Extreme Network OS command line interface (CLI), refer to the *Extreme Network OS Command Reference*.

What's new in this document

On October 30, 2017, Extreme Networks, Inc. acquired the data center networking business from Brocade Communications Systems, Inc. This document has been updated to remove or replace references to Brocade Communications, Inc. with Extreme Networks, Inc., as appropriate.

References to "fabric cluster mode" have been removed, as this is no longer supported.

This document supports the following features introduced in Network OS7.3.0:

- VLAG load balancing
- LACP for PXE

FCoE

- FCoE overview..... 13
- FCoE logical SAN overview..... 24
- Configuring FCoE..... 34

FCoE overview

Fibre Channel over Ethernet (FCoE) enables you to transport FC protocols and frames over Data Center Bridging (DCB) networks. DCB is an enhanced Ethernet network that enables the convergence of various applications in data centers (LAN, SAN, and HPC) onto a single interconnect technology.

FCoE provides a method of encapsulating the Fibre Channel (FC) traffic over a physical Ethernet link. FCoE frames use a unique EtherType [FCoE uses 0x8906 and FCoE Initialization Protocol (FIP) uses 0x8914] that enables FCoE SAN traffic and legacy LAN Ethernet traffic to be carried on the same link. FC frames are encapsulated in an Ethernet frame and sent from one FCoE-aware device across an Ethernet network to a second FCoE-aware device. The FCoE-aware devices may be FCoE end nodes (ENodes) such as servers, storage arrays, or tape drives on one end and FCoE Forwarders on the other end. FCoE Forwarders (FCFs) are switches providing SAN fabric services and may also provide FCoE-to-FC bridging.

The motivation behind using DCB networks as a transport mechanism for FC arises from the desire to simplify host protocol stacks and consolidate network interfaces in data center environments. FC standards allow for building highly reliable, high-performance fabrics for shared storage, and these characteristics are what DCB brings to data centers. Therefore, it is logical to consider transporting FC protocols over a reliable DCB network in such a way that it is completely transparent to the applications. The underlying DCB fabric is highly reliable and high performing, the same as the FC SAN.

In FCoE, ENodes discover FCFs and initialize the FCoE connection through the FCoE Initialization Protocol (FIP). FIP has a separate EtherType from FCoE. FIP includes a discovery phase in which ENodes discover VLANs supporting FCoE, solicit FCFs on those VLANs, and FCFs respond to the solicitations with advertisements of their own. At this point, the ENodes know enough about the FCFs to log in to them. The virtual link establishment and fabric login (FLOGI/FDISC) for VN-to-VF port connections is also part of FIP.

Network OS supports the following:

- 100-Gbps blades
- 40-Gbps breakout Inter-Switch Links (ISLs)
- Changes to the way in which the number of FCoE interfaces are created, through the **fcoe-enodes** command
- FCoE logical SANs
- FCoE troubleshooting commands

FCoE terminology

The following table lists and describes the FCoE terminology used in this document.

TABLE 1 FCoE terminology

Term	Description
FCoE	Fibre Channel over Ethernet
DCB	Data Center Bridging
VN_Port	FCoE equivalent of an FC N_Port

TABLE 1 FCoE terminology (continued)

Term	Description
VF_Port	FCoE equivalent of an FC F_Port
ENode	An FCoE device that supports FCoE VN_Ports (servers and target devices)

End-to-end FCoE

The Extreme VCS Fabric is a convergence-ready fabric. This means it is capable of providing lossless service and other features expected of a CEE-capable network. This includes support for multi-hop FCoE, where an FCoE initiator can communicate with an FCoE target that is a number of hops away.

FCoE operations

Each switch in the Extreme VCS Fabric cluster acts as a fully functional FCoE Forwarder (FCF). All Fibre Channel (FC) services required to support a Virtual Network (VN) must run on every Extreme VCS Fabric cluster switch, and each switch in the fabric acts as if it were a separate domain in an FC SAN.

For all practical purposes, an Extreme VCS Fabric operates similarly to an FC fabric because all the FCoE initiators and targets are connected to the Extreme VCS Fabric. Each switch in the cluster gets a domain ID, and once the fabric forms, all the FC services (such as Name Server, Login Controller, Domain Controller) are available on each individual cluster switch.

Network OS 4.0.0 and later supports FCR/LSAN zoning. A combination of 2000 FCoE devices and 1000 FC routed devices (for a total maximum of 3000 devices) is the fabric limit. Because open zoning floods all the State Change Notifications (SCNs) to every device, it should be used only when the fabric has 300 total devices or fewer. Fabrics with higher device counts should have user-defined zoning configurations, with a maximum of 255 devices per zone.

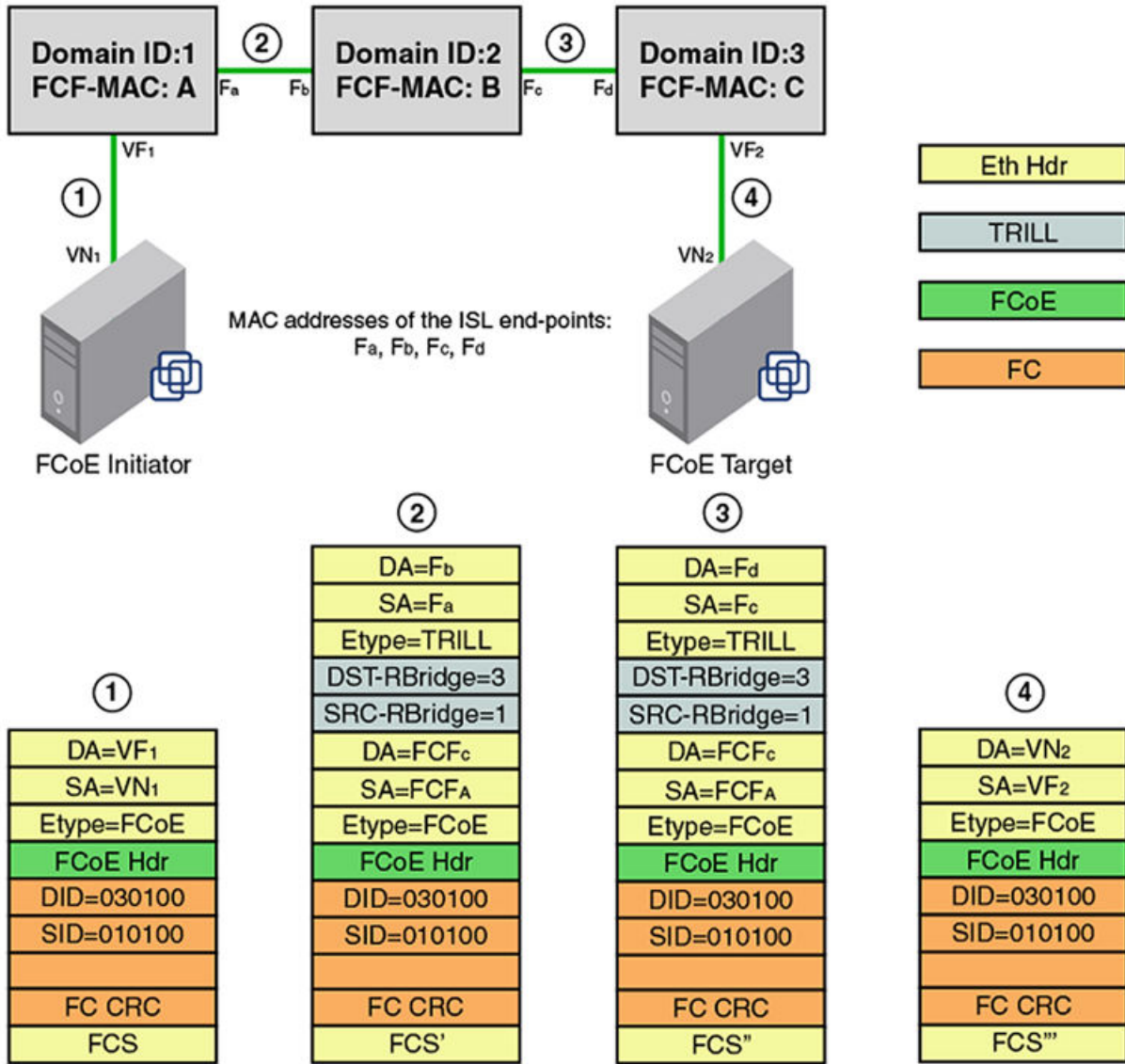
FCoE traffic forwarding across the fabric follows the same equal-cost multi-path (ECMP) routing rules as does LAN traffic forwarding.

FCoE end-to-end forwarding

FCoE frame forwarding between two FCoE devices attached to the Extreme VCS Fabric works similarly to Layer 3 IP routing. The end-node talks to the default gateway's MAC address and the Layer 2 headers are modified hop-by-hop until the frame reaches its final destination. Forwarding decisions are based on the contents of the IP header in the case of IP routing, and the IP header is untouched along the path. FCoE forwarding works the same way.

The following figure illustrates this process. Assume that VN1 (an FCoE initiator) is trying to access VN2 (an FCoE target).

FIGURE 1 FCoE end-to-end header process



1. VN1 and VN2 discover VF1 and VF2 through FIP Discovery Protocol and perform a Fabric Login (FLOGI) to their respective VF ports. That is, VN1 performs an FIP FLOGI to VF1 and VN2 performs a FIP FLOGI to VF2. This works like IP in that all communication between the end-station and the network happens to the router's MAC address at Layer 2. This means VN1 is always communicating with VF1 at Layer 2.
2. In an Extreme VCS Fabric implementation, all FC services are available on every cluster unit. This means there is Fibre Channel Network Switch (FCNS) available on both FCF1 and FCF2. The FCNS service functions identically as it does in an FC SAN. As a result, VN1 discovers VN2.

3. VN1 attempts an N_Port Login (PLOGI) to VN2, with the frame information shown at point 1 in the following figure. The Layer 2 header contains VF1 as the destination MAC address. The Layer 3 header (in this case, the FC header) contains the actual DID and SID of the initiator and the target respectively.

In this example, because VN1 is connected to the FCF with a Domain ID of 1, its PID is 010100. Similarly, because VN2 is connected to FCF3, its FC address is 030100.

4. When FCF-A receives the frame on VF1, it performs a Layer 3 lookup. It looks up the DID in the FC header and determines that the frame is destined to a non-local domain. FCF-A decodes the next hop needed to reach the destination domain of 3, based on Fabric Shortest Path First (FSPF). It is at this point that it does something different than a normal IP router.
5. FCF-A now knows that it needs to reach FCF-C. Each FCF in the Extreme VCS Fabric is assigned an FCF MAC address. FCF-A constructs the Layer 2 header based on this information. So, the original MAC header is now transformed as follows: the DA is changed from VF1 to FCF-C and the SA is changed from VN1 to FCF-A. This occurs at point 2 in the above figure.
6. The frame gets a Transparent Interconnection of Lots of Links (TRILL) header and traverses across the fabric to reach FCF-C. The TRILL header indicates that the source is RBridge 1 and the destination is RBridge 3. This occurs at point 2 in the above figure.
7. The outer MAC header is a link level header that gets the frame from FCF-A to FCF-B. FCF-B receives the frame. FCF-B scans the TRILL header, decodes the destination RBridge ID in the frame, and forwards the frame. FCF-B only modifies the Layer 2 header. It neither looks up nor modifies anything in the FC header or the inner MAC header. This occurs at point 3 in the above figure.
8. FCF-C receives the frame. FCF-C scans the TRILL header and decodes the destination RBridge ID. FCF-C promotes the frame to Layer 3 lookup, because the FCF-C is the DA in the inner MAC header. FCF-C then scans the FC header and does something similar to an area route lookup in FC SAN. This lookup yields the MAC address of VN2 and the VF interface (in this case, VF2) information that it needs to use to forward the frame to VN2. This occurs at point 4 in the above figure.
9. VN2 receives the PLOGI. The PLOGI response from VN2 traverses back to VN1 in similar fashion.

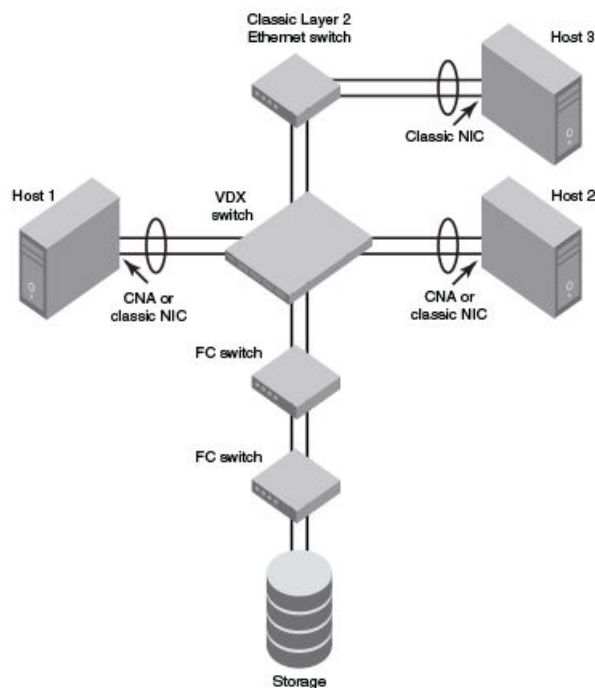
NOTE

It is assumed that both VN1 and VN2 are configured to be in the same FCoE VLAN, and FCoE forwarding is enabled on this VLAN in the Extreme VCS Fabric. Network OS v4.0.0 and later supports only one FCoE VLAN for all FCoE devices connected to the fabric.

FCoE and Layer 2 Ethernet

The Extreme VDX hardware contains DCB ports that support FCoE forwarding. The DCB ports are also backwards-compatible and support classic Layer 2 Ethernet networks (as shown in the following figure). In Layer 2 Ethernet operation, a host with a Converged Network Adapter (CNA) can be directly attached to a DCB port on the Extreme VDX hardware. Another host with a classic 10-gigabit Ethernet network interface card (NIC) can be either directly attached to a DCB port, or attached to a classic Layer 2 Ethernet network that is attached to the Extreme VDX hardware.

FIGURE 2 Multiple switch fabric configuration



Layer 2 forwarding

Layer 2 Ethernet frames are forwarded on the DCB ports. 802.1Q VLAN support is used to tag incoming frames to specific VLANs, and 802.3ac VLAN tagging support is used to accept VLAN tagged frames from external devices.

Network OS uses the following 802.1D bridging protocols between Layer 2 switches and to maintain a loop-free network environment:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)
- Per-VLAN Spanning Tree (PVST+)
- Rapid Per-VLAN Spanning Tree (RPVST+)

For detailed information on configuring these protocols, refer to [STP-Type Protocols](#) on page 87.

The Extreme VDX hardware handles Ethernet frames as follows:

- When the destination MAC address is not in the lookup table, the frame is flooded on all ports in the same VLAN, except the ingress port.
- When the destination MAC address is present in the lookup table, the frame is switched only to the correct egress port.
- When the destination MAC address is present in the lookup table, and the egress port is the same as the ingress port, the frame is dropped.
- If the Ethernet Frame Check Sequence (FCS) is incorrect, because the switch is in cut-through mode, a correctly formatted Ethernet frame is sent out with an incorrect FCS.
- If the Ethernet frame is too short, the frame is discarded and the error counter is incremented.
- If the Ethernet frame is too long, the frame is truncated and the error counter is incremented. The truncated frame is sent out with an incorrect FCS.

- Frames sent to a broadcast destination MAC address are flooded on all ports in the same VLAN, except the ingress port.
- When MAC address entries in the lookup table time out, they are removed. In this event, frame forwarding changes from unicast to flood.
- An existing MAC address entry in the lookup table is discarded when a device is moved to a new location. When a device is moved, the ingress frame from the new port causes the old lookup table entry to be discarded and the new entry to be inserted into the lookup table. Frame forwarding remains unicast to the new port.
- When the lookup table is full, new entries replace the oldest MAC addresses after the oldest MAC addresses reach a certain age and time out. MAC addresses that still have traffic running are not timed out.
- If the port is receiving jumbo frame packets and the port is not configured with the required MTU size to support jumbo frames, then the port discards those frames and increments the over-sized packet error counter.
- If the port is receiving valid multicast frames and the port is **not** part of a VLAN that is enabled for IGMP snooping, then the frames are treated as broadcast frames.
- If the port is receiving multicast frames with a destination MAC address (multicast MAC address) and destination IP address (multicast IP address) belonging to different group addresses or not pointing to the same group, the frames are silently discarded by the port.

NOTE

New entries start replacing older entries when the lookup table reaches 90 percent of its 32Kb capacity.

802.1Q VLAN tagging

The Layer 2 switch always tags an incoming frame with an 802.1Q VLAN ID. If the incoming frame is untagged, then a tag is added according to the port configuration. A port can classify untagged traffic to a single VLAN or to multiple VLANs. If the incoming frame is already tagged, then the port will either forward or discard the frame according to allowed VLAN rules in the port configuration.

These are three examples of 802.1Q VLAN tagging:

- If the DCB port is configured to tag incoming frames with a single VLAN ID, then incoming frames that are untagged are tagged with the VLAN ID.
- If the DCB port is configured to tag incoming frames with multiple VLAN IDs, then incoming frames that are untagged are tagged with the correct VLAN ID based on the port setting.
- If the DCB port is configured to accept externally tagged frames, then incoming frames that are tagged with a VLAN ID are passed through unchanged.

NOTE

Only a single switch-wide VLAN is capable of forwarding FCoE traffic.

For detailed information on configuring VLANs, refer to [802.1Q VLANs](#) on page 49.

Support for Virtual Fabrics

Network OS provides a Virtual Fabrics feature that supports multitenancy by extending the standard (802.1Q) VLAN ID space from 4096 through 8191, enabling the use of classified VLANs. Following an upgrade to Network OS 4.1, the system operates in native VLAN mode until the Virtual Fabrics feature is enabled. In this release, FCoE VLANs are limited to the 802.1Q range of 1 through 4096. FCoE frames are now able to accommodate 802.1AD S-TAGs (service provider tags) and C-TAGs (customer tags) for future support. A C-TAG used to classify an FCoE frame is the same as the VLAN ID and is system wide.

NOTE

Currently, FCoE VLANs can be only 802.1Q VLANs. They cannot be classified or used as C-TAGs for other VLAN classification. All tenant FCoE traffic rides on the same default FCoE VLAN (1002) as in the previous Network OS releases.

Incoming frame classification

The Extreme VDX hardware is capable of classifying incoming Ethernet frames based on the following criteria:

- Port number
- Protocol
- MAC address

The classified frames can be tagged with a VLAN ID or with 802.1p Ethernet priority. The 802.1p Ethernet priority tagging is done using the Layer 2 Class of Service (CoS). The 802.1p Ethernet priority is used to tag frames in a VLAN with a Layer 2 CoS to prioritize traffic in the VLAN. The Extreme VDX hardware also accepts frames that have been tagged by an external device.

Frame classification options are as follows:

- VLAN ID and Layer 2 CoS by physical port number — With this option, the port is set to classify incoming frames to a preset VLAN ID and the Layer 2 CoS on a physical port on the Extreme VDX hardware.
- VLAN ID and Layer 2 CoS by LAG virtual port number — With this option, the port is set to classify incoming frames to a preset VLAN ID and Layer 2 CoS on a Link Aggregation Group (LAG) virtual port.
- Layer 2 CoS mutation — With this option, the port is set to change the Layer 2 CoS setting by enabling the QoS mutation feature.
- Layer 2 CoS trust — With this option, the port is set to accept the Layer 2 CoS of incoming frames by enabling the QoS trust feature.

Congestion control and queuing

The Extreme VDX hardware supports several congestion control and queuing strategies. As an output queue approaches congestion, Random Early Detection (RED) is used to selectively and proactively drop frames to maintain maximum link utilization. Incoming frames are classified into priority queues based on the Layer 2 CoS setting of the incoming frame, or the possible rewriting of the Layer 2 CoS field based on the settings of the DCB port or VLAN.

The Extreme VDX hardware supports a combination of two scheduling strategies to queue frames to the egress ports: Priority queuing, which is also referred to as strict priority, and Deficit Weighted Round Robin (DWRR) queuing.

The scheduling algorithms work on the eight traffic classes as specified in 802.1Qaz Enhanced Transmission Selection (ETS).

Queuing features are described as follows:

- RED — RED increases link utilization. When multiple inbound TCP traffic streams are switched to the same outbound port, and some traffic streams send small frames while other traffic streams send large frames, link utilization will not be able to reach 100 percent. When RED is enabled, link utilization approaches 100 percent.
- Classification — Setting user priority.
 - Inbound frames — Inbound frames are tagged with the user priority set for the inbound port. The tag is visible when examining the frames on the outbound port. By default, all frames are tagged to priority zero.
 - Externally tagged Layer 2 frames — When the port is set to accept externally tagged Layer 2 frames, the user priority is set to the Layer 2 CoS of the inbound frames.
- Queuing
 - Input queuing — Input queuing optimizes the traffic flow in the following way. A DCB port has inbound traffic that is tagged with several priority values, and traffic from different priority settings is switched to different outbound ports. Some

outbound ports are already congested with background traffic while others are uncongested. With input queuing, the traffic rate of the traffic streams switched to uncongested ports should remain high.

- Output queuing — Output queuing optimizes the traffic flow in the following way. Several ports carry inbound traffic with different priority settings. Traffic from all ports is switched to the same outbound port. If the inbound ports have different traffic rates, some outbound priority groups will be congested while others can remain uncongested. With output queuing, the traffic rate of the traffic streams that are uncongested should remain high.
- Multicast rate limit — A typical multicast rate limiting example is where several ports carry multicast inbound traffic that is tagged with several priority values. Traffic with different priority settings is switched to different outbound ports. The multicast rate limit is set so that the total multicast traffic rate on output ports is less than the specified set rate limit. Multicast rate-limiting commands are not supported on the Extreme VDX 6740 or VDX 8770. On the latter platforms, use BUM storm control instead.
- Multicast input queuing — A typical multicast input queuing example is where several ports carry multicast inbound traffic that is tagged with several priority values. Traffic with different priority settings is switched to different outbound ports. Some outbound ports are already congested with background traffic while others are uncongested. The traffic rate of the traffic streams switched to the uncongested ports should remain high. All outbound ports should carry some multicast frames from all inbound ports. This enables multicast traffic distribution relative to the set threshold values.
- Multicast output queuing — A typical multicast output queuing example is where several ports carry multicast inbound traffic. Each port has a different priority setting. Traffic from all ports is switched to the same outbound port. If the inbound ports have varying traffic rates, some outbound priority groups will be congested while others remain uncongested. The traffic rate of the traffic streams that are uncongested remains high. The outbound ports should carry some multicast frames from all the inbound ports.
- Scheduling — A typical example of scheduling policy (using Strict Priority 0 and Strict Priority 1 modes) is where ports 0 through 7 carry inbound traffic, each port has a unique priority level, port 0 has priority 0, port 1 has priority 1, and so on. All traffic is switched to the same outbound port. In Strict Priority 0 mode, all ports have DWRR scheduling; therefore, the frames per second (FPS) on all ports should correspond to the DWRR settings. In Strict Priority 1 mode, priority 7 traffic uses Strict Priority; therefore, priority 7 can achieve a higher FPS. Frames from input ports with the same priority level should be scheduled in a round robin manner to the output port.

When setting the scheduling policy, each priority group that is using DWRR scheduling can be set to use a percentage of the total bandwidth by setting the PG_Percentage parameter.

Access control

Access Control Lists (ACLs) are used for Layer 2 switching security. Standard ACLs inspect the source address for the inbound ports. Extended ACLs provide filtering by source and destination addresses and protocol. ACLs can be applied to the DCB ports or to VLANs.

ACLs function as follows:

- A standard Ethernet ACL configured on a physical port is used to permit or deny frames based on the source MAC address. The default is to permit all frames.
- An extended Ethernet ACL configured on a physical port is used to permit or deny frames based on the source MAC address, destination MAC address, and EtherType. The default is to permit all frames.
- A standard Ethernet ACL configured on a LAG virtual port is used to permit or deny frames based on the source MAC address. The default is to permit all frames. LAG ACLs apply to all ports in the LAG.
- An extended Ethernet ACL configured on a LAG virtual port is used to permit or deny frames based on the source MAC address, destination MAC address, and EtherType. The default is to permit all frames. LAG ACLs apply to all ports in the LAG.
- A standard Ethernet ACL configured on a VLAN is used to permit or deny frames based on the source MAC address. The default is to permit all frames. VLAN ACLs apply to the Switched Virtual Interface (SVI) for the VLAN.

- An extended Ethernet ACL configured on a VLAN is used to permit or deny frames based on the source MAC address, destination MAC address, and EtherType. The default is to permit all frames. VLAN ACLs apply to the Switched Virtual Interface (SVI) for the VLAN.

For detailed information on configuring ACLs, refer to the "Configuring and Managing ACLs" section of the *Network OS Security Configuration Guide*.

Trunking

NOTE

The term "trunking" in an Ethernet network refers to the use of multiple network links (ports) in parallel to increase the link speed beyond the limits of any one single link or port, and to increase the redundancy for higher availability.

802.1ab Link Layer Discovery Protocol (LLDP) is used to detect links to connected switches or hosts. Trunks can then be configured between an adjacent switch or host and the Extreme VDX hardware.

The Data Center Bridging Capability Exchange Protocol (DCBX) extension is used to identify a DCB-capable port on an adjacent switch or host.

The 802.3ad Link Aggregation Control Protocol (LACP) is used to combine multiple links to create a trunk with the combined bandwidth of all the individual links. For detailed information on configuring LACP, refer to [Link Aggregation](#) on page 117.

Flow control

802.3x Ethernet pause and Ethernet Priority-based Flow Control (PFC) are used to prevent dropped frames by slowing traffic at the source end of a link. When a port on a switch or host is not ready to receive more traffic from the source, perhaps due to congestion, it sends pause frames to the source to pause the traffic flow. When the congestion has been cleared, it stops requesting the source to pause traffic flow, and traffic resumes without any frame drop.

NOTE

Ethernet pause differs from PFC in that the former is applied to all traffic streams irrespective of their COS values, whereas the latter is always applied to a specific COS or priority value.

When Ethernet pause is enabled, pause frames are sent to the traffic source. Similarly, when PFC is enabled, there is no frame drop; pause frames are sent to the source switch.

Support for 40-Gbps ISLs on breakout ports

40-gigabit-per-second ISLs are supported, including on breakout ports.

FCoE Initialization Protocol

The FCoE Initialization Protocol (FIP) discovers and establishes virtual links between FCoE-capable entities connected to an Ethernet cloud through a dedicated EtherType (0x8914) in the Ethernet frame.

FIP discovery

NOTE

ANSI INCITS 462-2010 Fibre Channel - Backbone - 5 (FC-BB-5) / 13-May-2010 is supported.

The Extreme VDX hardware FIP discovery phase operates as follows:

- The Extreme VDX hardware uses the FCoE Initialization Protocol (FIP). ENodes discover VLANs supporting FCoE, FCFs, and then initialize the FCoE connection through the FIP.
- VF_Port configuration — An FCoE port accepts ENode requests when it is configured as a VF_Port and enabled. An FCoE port does not accept ENode requests when disabled.
- Solicited advertisements — A typical scenario is where an Extreme VDX hardware receives a FIP solicitation from an ENode. Replies to the original FIP solicitation are sent to the MAC address embedded in the original FIP solicitation. After being accepted, the ENode is added to the VN_Port table.
- VLAN 1 — The Extreme VDX hardware should not forward FIP frames on VLAN 1 because it is reserved for management traffic only.
- A fabric-provided MAC address is supported.

NOTE

In the fabric-provided MAC address format, VN_Port MAC addresses are based on a 48-bit fabric-supplied value. The first three bytes of this value are referred to as the FCMAP. The next three bytes are the FC ID, which is assigned by the switch when the ENode logs in to the switch.

FIP login

FIP login operates as follows:

- ENodes can log in to the Extreme VDX hardware using FIP, Fabric login (FLOGI) or fabric discovery (FDISC).
- Extreme VDX hardware in the fabric maintains the MAC address, World Wide Name (WWN), and PID mappings per login. Each ENode port should have a unique MAC address and WWN.
- FIP FLOGI — The Extreme VDX hardware accepts the FIP FLOGI from the ENode. The FIP FLOGI acceptance (ACC) is sent to the ENode if the ENode MAC address or WWN matches the VN_Port table on the Extreme VDX hardware. The FIP FLOGI request is rejected if the ENode MAC address or WWN does not match. The ENode login is added to the VN_Port table. Fabric Provided MAC Addressing (FPMA) is supported.
- FIP FDISC — The Extreme VDX hardware accepts FIP FDISC from the ENode. FIP FDISC acceptance (ACC) is sent to the ENode if the ENode MAC address or WWN matches the VN_Port table on the Extreme VDX hardware. The FIP FDISC request is rejected if the ENode MAC address or WWN does not match. The ENode login is added to the VN_Port table. FPMA is supported.
- Maximum logins per VF_Port (logical FCoE port) — The Extreme VDX hardware supports a maximum of 64 logins. The VF_Port rejects further logins after the maximum is reached.
- Maximum logins per switch — The Extreme VDX hardware accepts a maximum of 1000 logins per switch.
- Maximum logins per VCS cluster — 3000.

FIP logout

FIP logout operates as follows:

- ENodes and VN_Ports can log out from the Extreme VDX hardware using FIP. The Extreme VDX hardware in the fabric updates the MAC address, WWN, and PID mappings upon logout. The Extreme VDX hardware also handles scenarios of implicit logout where the ENode has left the fabric without explicitly logging out.
- FIP logout (LOGO) — The Extreme VDX hardware accepts a FIP LOGO from the ENode. The FIP LOGO acceptance (ACC) should be sent to the ENode if the ENode MAC address and the VN_Port MAC address matches the VN_Port table data on the switch. The LOGO is ignored (not rejected) if the ENode MAC address does not match. The ENode logout is updated in the VN_Port table.

- Implicit logout — With the ENode directly connected to a DCB port, if the port that the ENode is attached to goes offline, the Extreme VDX hardware implicitly logs out that ENode. ENode logout is updated in the VN_Port table. The Extreme VDX hardware sends an FIP Clear Virtual Links (CVL) to the ENode.

The FIP Virtual Link Maintenance protocols provide a mechanism to detect reachability loss to an ENode or any VN_Port instantiated on that ENode. This is accomplished by the periodic transmission of FIP Keep-Alive (FKA) messages from the ENode.

If FKA timeouts are enabled on the switch, all VN_Ports associated with an ENode will be implicitly logged out in the event of an ENode FKA timeout.

If FKA timeouts are enabled on the switch, the VN_Port will be implicitly logged out in the event of a VN_Port FKA timeout.

Name server operation

The Extreme VDX hardware name server function operates as follows:

- ENode login and logout to and from the Extreme VDX hardware updates the name server in the FC fabric. The Extreme VDX hardware maintains the MAC address to WWN and PID mappings.
- ENode login and logout — When an ENode login occurs through any means (FIP FLOGI, FIP FDISC, FCoE FLOGI, or FCoE FDISC), an entry is added to the name server. When an ENode logout occurs through any means (FIP LOGO, FCoE LOGO, or implicit logout), the entry is removed from the name server.
- ENode data — The Extreme VDX hardware maintains a VN_Port table. The table tracks the ENode MAC address, FIP login parameters for each login from the same ENode, and WWN and PID mappings on the FC side. You can display the VN_Port table with the **show fcoe login** command.

Registered State Change Notification

The Extreme VDX hardware Registered State Change Notification (RSCN) function operates as follows:

- RSCN events generated in the FC fabric are forwarded to the ENodes. RSCN events generated on the FCoE side are forwarded to the FC devices. DCB is not aware of RSCN events.
- Device RSCN — An RSCN is generated to all registered and affected members when an ENode either logs in or logs out of an FCF through any means. An RSCN is generated when an FC N_Port device either logs in or logs out of the FC fabric.

NOTE

When transmitting an RSCN, zoning rules still apply for FCoE devices as the devices are treated as regular FC N_Ports.

- VF_Port RSCN — An RSCN is generated to all registered members when a VF_Port goes online or offline, causing ENode or FC devices to be added or removed.
- Domain RSCN — An RSCN is generated to all registered and affected members when an FC switch port goes online or offline, causing ENode or FC devices to be added or removed. An RSCN is generated when two FC switches merge or segment, causing ENode or FC devices to be added or removed. When FC switches merge or segment, an RSCN is propagated to ENodes.
- Zoning RSCN — An RSCN is generated to all registered and affected members when a zoning exchange occurs in the FC fabric.

Local ENode configuration

The number of interfaces to be created is configured per switch by means of the **fcoe-enodes** command, executed in RBridge ID configuration mode. This is known as the local ENode configuration model. The number of ENodes that can be configured ranges from 0 through 1000, with a default of 64.

An FCoE license is required to enable FCoE interfaces. If this license is not present, no FCoE interfaces are created.

FCoE queuing

The QoS configuration controls the FCoE traffic distribution.

NOTE

Changing these settings requires changes on both the Extreme VDX hardware and the Converged Network Adapter (CNA); therefore, the link must be taken offline and put back online after a change is made.

Traffic scheduler configuration changes affect FCoE traffic distribution as follows:

- Changing the priority group for a port causes the FCoE traffic distribution to be updated. The priority group and bandwidth are updated.
- Changing the priority table for a port causes the FCoE traffic distribution to be updated. The CoS-to-priority group mapping is updated.
- Changing the class map for a port causes the FCoE traffic distribution to be updated.
- Changing the policy map for a port causes FCoE traffic distribution to be updated.
- Changing the DCB map for a port causes the FCoE traffic distribution to be updated.
- The FCMAP-to-VLAN mapping determines the FCoE VLAN allowed for the FCoE session. Modifying this mapping causes the existing sessions to terminate.

NOTE

Only one FCoE VLAN is supported in Network OS 4.0.0 and later releases.

FCoE logical SAN overview

The FCoE logical SAN feature supports up to four logical storage area networks in a VCS Fabric, in addition to the default FCoE VLAN SAN support. All nodes in a VCS Fabric must be upgraded to Network OS 6.0.0 or 6.0.1 for multiple FCoE logical SAN support.

A logical SAN provides Fibre Channel SAN connectivity to the FCoE device on any node in a VCS Fabric through an Access Gateway (AG) in the fabric or through a node with an E_Port configured to connect to an Extreme Fibre Channel Routing (FCR) EX_Port in a Fibre Channel SAN. Logical SANs can be configured as either local or remote SANs. A local logical SAN provides logical separation within the VCS Fabric, while a remote logical SAN provides FCoE SAN connectivity through an AG. Each AG can provide connectivity to only one remote logical SAN. Other switches in the VCS Fabric can be configured to support multiple logical SANs apart from the default SAN. Each logical SAN, identified by the fabric map configuration, uses a separate VLAN to provide traffic isolation.

For the local logical SANs within the VCS Fabric, the name server and Fabric Shortest Path First (FSPF) continue to treat all the SANs as a single fabric. On a local logical SAN, logins are serviced by the switch that is directly connected to the FCoE device.

For the default SAN, logins are serviced by the switch that is directly connected to the FCoE device. For the remote logical SANs, logins are serviced by the AG that is connected to the Fibre Channel SAN. The AG must be configured as an FCoE Forwarder (FCF) for that remote logical SAN. The non-AG switch, which is connected to the network by means of an FCoE converged network adapter (CNA) and acts only as a pass-through switch that sends traffic to the AG, is called an FCoE Initialization Protocol (FIP) forwarder, or FIF. The FIF-

to-FCF mapping, whereby the FIF communicates with the FCF in the VCS Fabric, is determined by the FCF group configuration that they are part of. Intermediate switches that are connected between the FCF and the FIF need no special configuration.

The initial release of this feature (Network OS 6.0.0) supported four remote logical SANs within a single VCS Fabric, with each FIF supporting a default FCoE VLAN SAN and a remote logical SAN. Now, Beginning with Network OS 6.0.1, in addition to the default FCoE VLAN SAN, the local logical SAN feature is provided. In a VCS Fabric, the maximum logical SAN support is four logical SANs. These four logical SANs can be all local, all remote, or a combinations of the two. Also starting Network OS 6.0.1, the FCoE port profile is enhanced to support logical SAN configuration.. it is now possible to map multiple local FCoE ports to a specified VLAN and fabric map, as well as to specified RBridge IDs, by means of the **fcport-group** command and the associated **fcport-group-rbid** command.

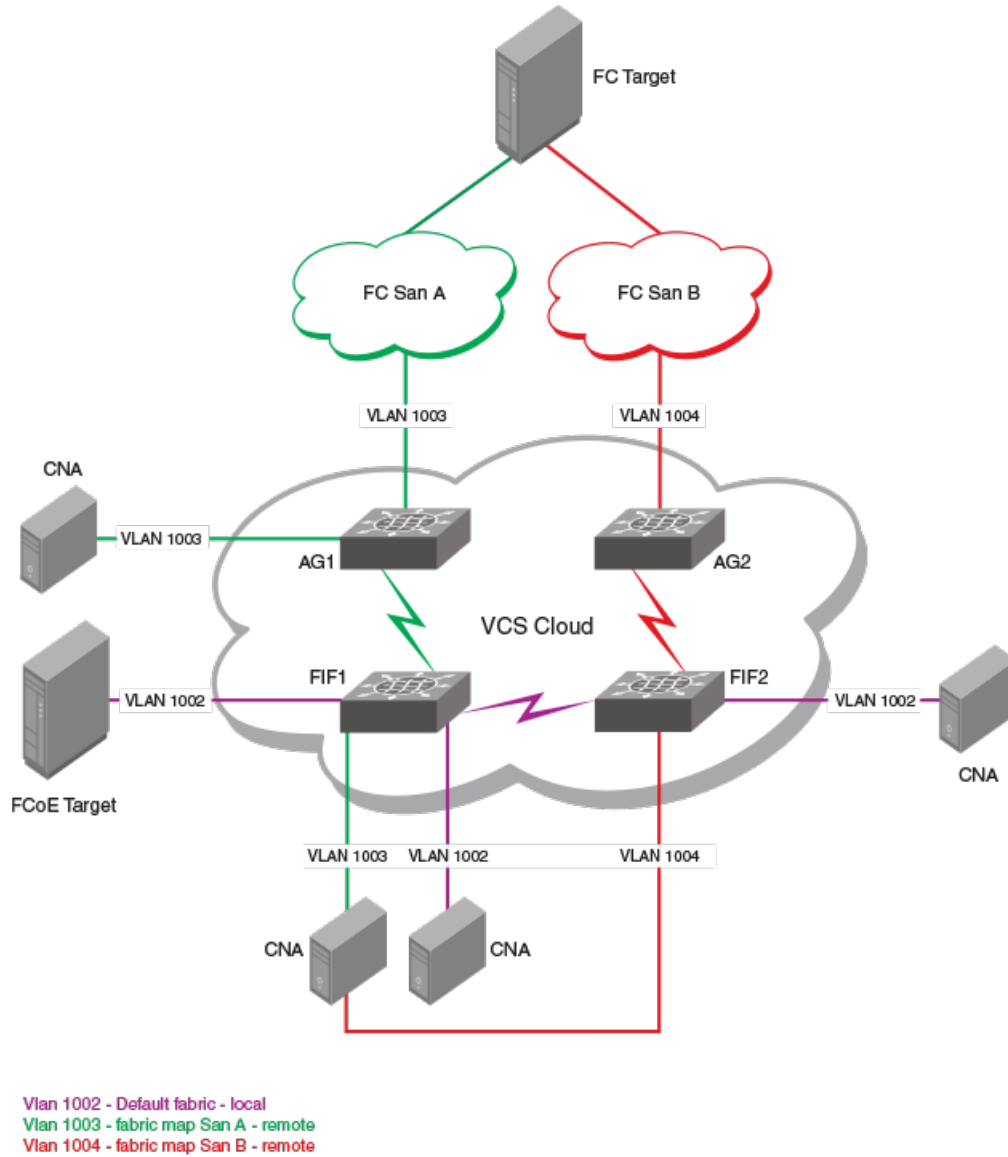
FCoE logical SAN use cases

The following topologies illustrate the three supported use cases.

FCoE logical SAN use case 1

In this use case, each FIF can support a default FCoE VLAN SAN and a remote logical SAN, as illustrated in the following figure. VLAN 1002 is local and supports the default fabric. VLANs 1003 and 1004 are remote and support SANs A and B, respectively.

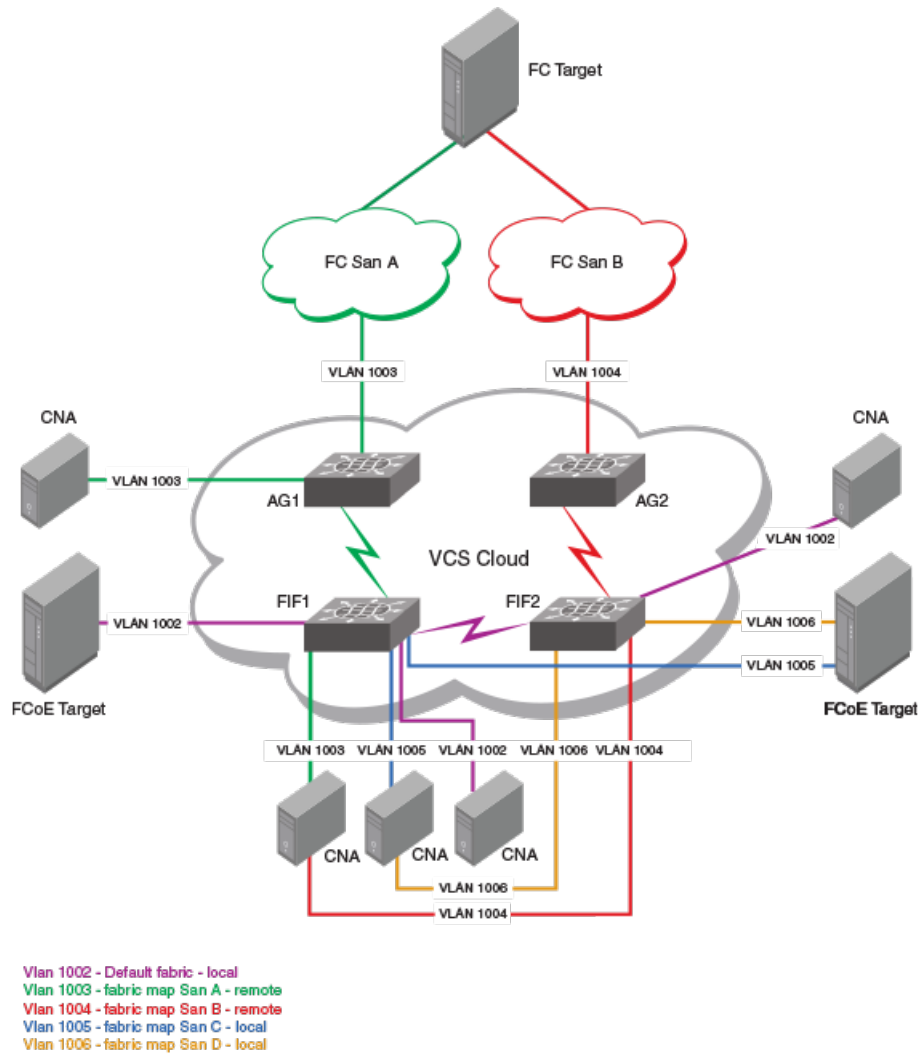
FIGURE 3 Use case 1: One remote SAN and a default FCoE VLAN SAN with a single FIF



FCoE logical SAN use case 2

In this use case, illustrated below, one remote logical SAN and one local logical SAN are allowed on a single FIF.

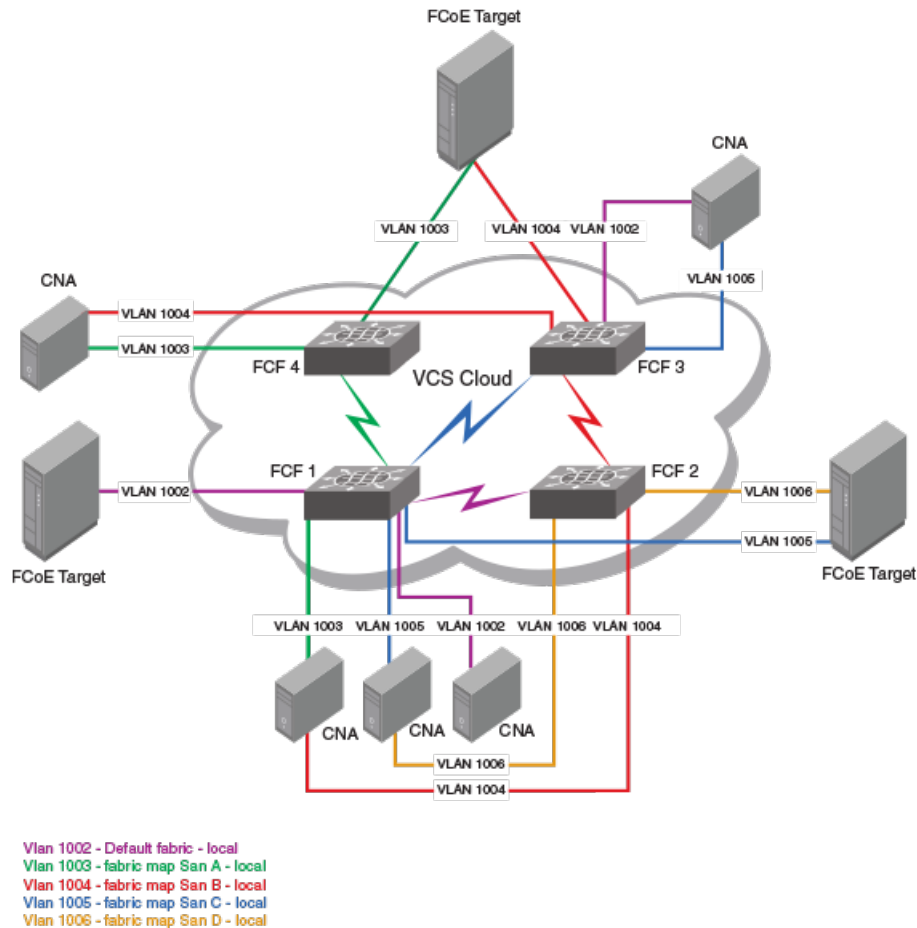
FIGURE 4 Use case 2: One remote logical SAN and one local logical SAN with a single FIF



FCoE logical SAN use case 3

In this use case, illustrated below, four logical SANs are supported. Each FIF supports two local logical SANs in addition to the default FCoE VLAN SAN. Local SANs do not connect to the FC SAN through the Access Gateway.

FIGURE 5 Use case 3: Four logical SANs



FCoE logical SAN behavior and provisioning model

When a VDX switch boots up in native FCoE mode (non-AG mode), its behavior is similar to the current local FCoE Forwarder (FCF) mode. The switch continues to support fcoepoint configurations on the local FCoE VLAN as before, by means of the default configuration.

Logical SAN behavior and provisioning

The following summarizes the behavior and provisioning of FCoE logical SANs:

- Every logical SAN is represented by a fabric map, and has a unique FCoE VLAN ID and FCoE MAC address prefix (FCMAP). The fabric map must be identified as either "local" or "remote."
- Every remote logical SAN is represented by an FCF group. Each FCF group must have one RBridge as an FCF to service the remote logical SAN and can have multiple RBridges as FIFs to service the remote logical SAN.
- Local logical SANs are available for provisioning on all RBridges in the fabric. FC ports in an RBridge can be provisioned into a local logical SAN by means of the **fcpport-group** command for the fabric map.

NOTE

FC ports provisioned in a local logical SAN can be in E_Port mode connecting to the EX end of an FC SAN through FCR, or can be in F_Port mode connecting to an FC end device.

- Ethernet ports can be provisioned for either local or remote logical SANs by means of the respective fabric maps for FCoE provisioning.

The following summarizes the configuration process:

- The user creates a fabric map by means of the **fabric-map** command, which in turn does the following automatically: **san-mode** is set to **remote**, one of the available VLANs is selected, and one of the available FCMAPs from the range 0E:FC:00 through 0E:FC:FF is selected. The user can change the VLAN and the FCMAP.
- Under the **fabric-map** command, the user uses the **fcf-group** command to create a group name and enter FCF group configuration mode. This mode allows the user to configure values for FCF and FIF RBridge IDs, set by the keywords **fcf-rbid** and **fif-rbid**, respectively. (A default fabric map does not contain an FCF group.)
- When the user attempts to configure an FCF RBridge, this is allowed only if the RBridge is enabled as an Access Gateway.
- In the case of fabric maps for local logical SANs, the user sets **san-mode** to **local** and uses the **fcport-group** command and the **fcport-group-id** subcommand to add RBridge IDs as appropriate. An "fcport-group" configuration can be part of a local fabric map only, and all FC ports in an attached RBridge are part of the local logical SAN.
- FCoE provisioning on interfaces is as before. The user uses the **fcoeport** command to configure physical interfaces for either local or remote logical SANs, as well as for the default FCoE logical SAN.
- The user applies FCoE provisioning on multiple ports by means of port profiles. The FCoE configuration is added to the FCoE subprofile of a port profile, and this profile is applied to ports as appropriate. (The FCoE subprofile configuration is applied like a VLAN subprofile and does not trigger MAC address learning on an interface.)
- The user associates each logical SAN to a port-profile domain, with each domain containing only one profile with the FCoE configuration. (Nondefault port profiles can have FCoE provisioning for only the nondefault fabric map.)
- Port-profile domains are configured only in Virtual Fabrics-enabled mode, and therefore port-profile support for logical SANs is provided only if Virtual Fabrics is enabled for all participating switches.

NOTE

FCoE provisioning for a logical SAN is not allowed on an Ethernet interface if the RBridge on which the interface is present is not part of the FCF group configuration, whether as an FCF RBridge ID or an FIF RBridge ID.

ATTENTION

The FCoE SAN configuration is global, and therefore must be done on the principal switch in logical chassis cluster mode.

Understanding port profiles and multiple fabric maps

Although an FCoE configuration that is done as part of a profile is applied during the configuration and not as a result of MAC address learning, port-profile configuration allows the FCoE configuration to be aligned with the concept of domains. This provides a single point of configuration for all the ports that would form a connecting point in the VCS Fabric for a virtual machine (VM). (These are all ports to which a VM could be moved.) All ports that would support a particular VM must be configured with the fcoeport configuration for the appropriate SAN, so that the VM could boot from that FCoE SAN. This is achieved by mapping the logical SAN to one or more domains, so that each domain can have an FCoE configuration for only a particular SAN.

Prior to Network OS 6.0.0, FCoE profiles were supported only under the default port profile. Now the FCoE profile is supported under nondefault port profiles as well. Now the default port profile is provided only for backward compatibility, as well as for use cases in which the user wants to retain the default SAN and not use a logical SAN.

ATTENTION

It is recommended that only the domain-based application of port profiles (by means of the `port-profile-port` domain command) be used. A VCS Fabric can have port profiles with the default `fcoe-map` or the nondefault `fcoe-map`, but not both.

AMPP configurations

There are two Auto Migrating Port Profile (AMPP) configurations: global and interface level. These are discussed below with respect to the FCoE logical SANs feature.

Global port profiles

Global port profiles are configured by the `port-profile` command. The FCoE profile is configured by the `fcoeport` command, with the fabric map defined by the `fabric-map-name` variable. Nondefault maps are now supported. However, as these maps can be applied only through domains, Virtual Fabrics must be enabled to configure port profiles for nondefault SANs. As an FCoE port profile is applied during configuration and is not associated with any MAC address that must be learned, it is recommended that FCoE port profiles be kept separate from other port profiles.

The following is an example global port-profile configuration.

```
device(config)# port-profile A
device(config-port-profile-A)# fcoe-profile
device(config-fcoe-profile)# fcoeport sanA

device(config)# port-profile B
device(config-port-profile-B)# fcoe-profile
device(config-fcoe-profile)# fcoeport sanB

device(config)# port-profile C
device(config-port-profile-C)# fcoe-profile
device(config-fcoe-profile)# fcoeport sanA
```

NOTE

The default port profile can contain only the default FCoE map, and nondefault port profiles cannot contain the default FCoE map. Port-profile domains can contain many port profiles. However, only one port profile with an FCoE subprofile can be part of a domain. The same port profile can be part of multiple domains.

The following is an example global port-profile domain configuration.

```
device(config)# port-profile-domain D1
device(config-port-profile-domain-D1)# port-profile A
device(config-port-profile-domain-D1)# port-profile D

device(config)# port-profile-domain D2
device(config-port-profile-domain-D2)# port-profile A
device(config-port-profile-domain-D2)# port-profile E
```

Interface-level port profiles

Interface-level port profiles are configured by the `port-profile-port` command. The default FCoE port fabric map is automatically applied on an interface for default profiles where FCoE logical SANs are not enabled and the user wants to maintain a configuration created prior to Network OS 6.0.1. In previous releases, the default port profile is applied on all port-profile ports, irrespective of the port-profile domain. This behavior continues unless nondefault port profiles are configured.

The fabric map configured by the `fcoeport` command is automatically applied on an interface where `fabric-map-name` is the fabric map under the profile with the FCoE subprofile that is part of the domain.

The following are example interface-level port-profile configurations.

Three logical SANs:

```
fabric-map sana
fabric-map sanb
fabric-map sanc
```

Three FCoE port profiles:

```
Port-profile ProfileSanA
  fcoeport sana
Port-profile ProfileSanB
  fcoeport sanb
Port-profile ProfileSanC
  fcoeport sanc
```

Five port-profile domains:

NOTE

These support various VMs identified by MAC address, and are applied when the associated MAC addresses are learned.

```
Port-profile-domain DD1
  Port-profile ProfileSanA
  Port-profile PP1
  Port-profile PP2
  Port-profile PP3

Port-profile-domain DD2
  Port-profile ProfileSanB
  Port-profile PP1
  Port-profile PP4
  Port-profile PP5

Port-profile-domain DD3
  Port-profile ProfileSanC
  Port-profile PP6
  Port-profile PP7

Port-profile-domain DD4
  Port-profile ProfileSanA
  Port-profile PP5
  Port-profile PP3
  Port-profile PP7

Port-profile-domain DD5
  Port-profile ProfileSanB
  Port-profile PP4
  Port-profile PP6
```

Restrictions for port profiles and fabric maps

Note the following restrictions. This behavior is enforced to ensure that configurations are consistent and undesired combinations of configurations are avoided.

- Default configurations are not allowed on nondefault port profiles.
- Only one port profile with an FCoE subprofile is allowed in a port-profile domain.
- Changing a fabric map or an FCF group is not allowed if the fabric map contains a port-profile-port configuration.
- FCoE configuration by means of the **fcoeport** and **port-profile-port** commands can coexist, but only if the FCoE map configurations that are inside the port-profile domain and also are applied directly on an interface are not in conflict.
- Changing an FCoE port profile is not allowed when either (a) the port profile is activated or (b) the FCoE map conflicts with the map applied on any interface.

- Adding a port profile to a port-profile domain is not allowed when either (a) the port-profile domain already has an FCoE map or (b) the port-profile domain is applied on at least one interface for which there is an conflicting FCoE map.

FCoE logical SAN limitations

Note the following limitations for this feature:

- One Access Gateway (AG) can service only one FCoE VLAN and one remote logical SAN.
- If the AG is not part of any remote logical SAN, the default logical SAN (fabric map) on the AG acts as a remote logical SAN. Therefore, it is not part of the default logical SAN (fabric map) on other Extreme VDX series platforms that are not AGs. The default logical SAN (fabric map) on non-AG VDX platforms acts as a local logical SAN, and all FCoE devices logged into the default local logical SAN on non-AG VDX platforms can see each other.
- An FIF at the first hop can be part of only one AG.
- An FIF or FCF can be part of only one FCF group.
- An FIF can support only two logical SANs (and a maximum of one remote logical SAN) in addition to the default.
- A maximum of four logical SANs (local, remote, or a combination of the two) are allowed in a single VCS Fabric.
- Each node in the VCS Fabric can support four local logical SANs.
- Fabric Services share the same device namespace between local logical SANs. Therefore, zoning needs to be used to isolate devices within a local logical SAN.
- FC ports (Flexports) in non-AG VDXs are by default part of the default logical SAN (fabric map). These ports can be made part of any other logical SAN by means of the **fcport-group** command; the fcport-group configuration can be part of only one local fabric-map. All FCoE ports on non-AG VDXs can be part of only one local logical SAN. Only RBridge IDs can be part of the fcport-group configuration, and all FC ports on a given RBridge are part of the logical SAN.
- FC ports in non-AG VDXs are part of the remote logical SAN that the AG is part of. If the AG is not made part of any remote logical SAN, those ports are part of the default logical SAN, which acts as a remote logical SAN.
- Duplicate WWN detection is done across all local logical SANs. Duplicate WWNs are not detected across local and remote or between two remote logical SANs in a VCS Fabric. The duplicate WWN detection for a remote logical SAN must be done at the upstream switch.

The following features and platforms are not supported:

- Pre-FIP versions of CNAs
- FIP version 0
- Hard zoning within a VCS Fabric
- FCoE in standalone mode (that is, VCS is disabled)
- FCoE over VLAG and Extreme LAG
- Traffic isolation (By default, the entire VCS Fabric network is treated as a single fabric. All logical SAN frames use the shorted VCS path available to reach the switch to which the frames are destined. Intermediate switches in the VCS Fabric are just pass-through devices and hence play the same role as they currently do.)

For additional details related to this feature, see [FCoE logical SAN behavior and provisioning model](#) on page 28.

FCoE logical SAN upgrade/downgrade considerations

This section details the upgrade and downgrade considerations for this feature, including considerations for mixed-fabric deployments.

Upgrade considerations: NOS 5.0.x to NOS 6.0.x

- Non-ISSU upgrades from Network OS 5.0.x to Network OS 6.0.1 are supported.
- Following an upgrade, all existing FCoE configurations, including FCoE provisioning configurations on the Access Gateway, continue to be supported.
- The AG has the same functionality it had in Network OS 5.0.x.
- To use the new FCoE logical SAN features, the user must configure the required fabric maps and FCF groups for the logical SANs and accordingly change the FCoE interface provisioning on the AG.

Downgrade considerations: NOS 6.0.x to NOS 5.0.x


Non-ISSU downgrades from Network OS to Network OS 5.0.x are allowed only when there are no FCoE logical SAN configurations.

Upgrade considerations: NOS 6.0.0 to NOS 6.0.1

- All FCoE login sessions are restored after the upgrade. Devices re-login because the system goes through a cold recovery.
- To use the local logical SAN feature, the provisioning model must be followed to migrate from the default/remote FCoE VLAN to a new local FCoE VLAN, and configurations must be changed as appropriate.

Downgrade considerations: NOS 6.0.1 to NOS 6.0.0

- If the switch is configured with the local logical SAN feature, a downgrade is prevented.
- If FCoE is configured on any nondefault port profiles, the downgrade is prevented.

-  **CAUTION**
A downgrade is disruptive.

FCoE logical SAN scalability

The default limit is 64 ENodes. The following table lists the maximum limits for this feature.

TABLE 2 Maximum limits for FCoE logical SANs

Parameter	Limit
Virtual Fabrics (VF) ports per switch	1000
N_Port ID Virtualization (NPIV) instances per port	64
FCoE devices per VCS Fabric (FLOGI + FDISC)	2000
Total SAN devices per VCS Fabric	3000
LLDP neighbors per switch	VDX 8770: 384
	VDX 67xx: 60
	VDX 6940: 112
Total number of logical SANs	4, plus 1 default SAN
Number of default ENodes	64

FCoE logical SAN configuration recommendations

Note the following guidelines, which apply to changing both default and nondefault parameters:

- Configuration changes are not allowed when logins are present.
- Interfaces cannot be provisioned for FCoE (especially those for logical SANs) if the entire configuration for a logical SAN is not in place. All fabric map and FCF group configurations must be complete.
- The deletion of any of the following logical SAN components is not allowed when there are ports provisioned for a logical SAN: the fabric map or FCF group, the FCF RBridge ID, or the FIF RBridge ID.

Configuring FCoE

This section presents the tasks necessary to configure FCoE interfaces and logical SANs.

Configuring logical FCoE ports

When the switch boots, a pool of 64 FCoE ports is created. These ports are not bound to any physical ports. The bindings are created when an FLOGI is received on the switch. Any free port that is available from the pool is selected and bound to the physical port where the FLOGI is received. The default number of logical ports is 64, and the range of valid values is from 0 through 1000.

NOTE

Extreme VDX switches support FCoE multi-hops for as many as nine hops.

When the FCoE logical port is automatically bound to a 10-gigabit Ethernet LAG port, this is referred to as *dynamic binding*. This binding is valid only until the FLOGI session is valid. The binding is automatically removed when CNA logs out. To create a persistent binding between the logical FCoE port and an interface that can be used for static binding (FortyGigabitEthernet, HundredGigabitEthernet, Port-channel, TenGigabitEthernet, mac-address), use the **bind** command. This is stored in the configuration and retained across reboots.

NOTE

Only one type of binding can be used for each physical port, so the LAG binding configurations will overwrite each other.

To create additional logical FCoE ports, perform the following steps in RBridge ID configuration mode.

1. Enter FCoE configuration mode on an RBridge.

```
device(config-rbridge-id-1)# fcoe
device(config-rbridge-fcoe)#
```

2. Enter the **fcoe-enodes** command to set the maximum number of logins allowed on the switch.

```
device(config-rbridge-fcoe)# fcoe-enodes 384
```

To bind the logical FCoE ports to a physical port, perform the following steps:

3. Enter interface subtype configuration mode on the RBridge.

```
device(config-rbridge-id-1)# interface fcoe 1/1/55
device(config-if-fcoe-1/1/55)#
```

4. Bind the logical port to the physical port.

```
device(conf-if-fcoe-1/1/55)# bind tengigabitethernet 1/0/1
```

- In privileged EXEC mode, verify the FCoE ENode configuration, by using the **show fcoe fcoe-enodes** command.

```
device# show fcoe fcoe-enodes
=====
Rbridge-id      Fcoe-enodes
=====
1                384
2                64 [D]
6                0
```

Configuring fabric maps

Fabric maps are used to configure FCoE properties on interfaces.

NOTE

This is not supported for remote or local FCoE logical SANs.

A fabric map is a placeholder for an FCoE VLAN and an FCMAP.

A fabric map with the name "default" is created during system boot-up. The user is not allowed to delete or rename this map. By default, the FCoE VLAN associated with the fabric map is the native FCoE VLAN 1002, the 802.1Q priority associated with the fabric map is 3, and the associated FCMAP is 0E:FC:00. The user can modify the VLAN, the priority, or the FCMAP, but cannot delete any of these.

NOTE

A fabric map can be edited even if it is associated with an interface. However, the VLAN of the fabric map cannot be edited. All other parameters, such as the FCMAP, priority, and advertisement interval, can be modified.

Do the following to edit the parameters of the default fabric map.

- In global configuration mode, enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

- Enable the default fabric map and modify it as in the following example.

```
device(config-fcoe)# fabric-map default
device(config-fcoe-fabric-map-default)# vlan 1003
device(config-fcoe-fabric-map-default)# priority 4
device(config-fcoe-fabric-map-default)# fcmmap 0e:fc:11
```

NOTE

Extreme does not support non-FCoE traffic over an FCoE VLAN. The FCoE VLAN should not carry any mixed traffic.

Configuring FCoE logical SANs

This section presents the tasks required to configure FCoE logical SANs and manage those configurations. Refer to [FCoE logical SAN use case 2](#) on page 26 for this example.

The VLANs are assigned as follows (if they have not already been assigned).

NOTE

The first free unprovisioned VLANs are assigned to the fabric map.

VLAN	Fabric	Logical SAN (san-mode)
1002	Default	Local
1003	fabric map SanA	Remote
1004	fabric map SanB	Remote
1005	fabric map SanC	Local
1006	fabric map SanD	Local

The configuration is global and consists of the following:

1. Configuring the FCoE logical SAN fabric maps to identify the nondefault SANs
2. For remote logical SANs, configuring the FCoE logical SAN FCF groups to map the FIFs to FCFs within a fabric map of nondefault SANs
3. Configuring the Access Gateways as FCoE Forwarder (FCF) RBridge IDs and their participation in the nondefault SANs

NOTE

It is also possible to map multiple local FCoE ports to a specified VLAN and fabric map, as well as to specified RBridge IDs,

For the configurations of the remaining use cases, refer to [FCoE logical SANs configuration examples](#) on page 44.

Creating fabric maps for logical SANs

This task creates fabric maps for a nondefault SAN.

Do the following to create and configure a fabric map for nondefault SAN SanA.

1. In global configuration mode, enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

2. Create the fabric map instance and enter FCoE fabric-map configuration mode.

```
device(config-fcoe)# fabric-map SanA
device(config-fcoe-fabric-map-SanA)#
```

NOTE

Values for VLAN, priority, and FCMAP are selected automatically from available values.

3. Change the VLAN from the default.

```
device(config-fcoe-fabric-map-SanA)# vlan 1003
```

4. Change the priority from the default.

```
device(config-fcoe-fabric-map-SanA)# priority 3
```

NOTE

The priority should be the same for all fabric maps.

- Change the fabric-provided MAC address (FPMA) FCoE MAC address prefix (FCMAP) from the default.

```
device(config-fcoe-fabric-map-SanA)# fcmmap 0e:fc:10
```

NOTE

With multiple fabric maps, each has its own FCMAP value. Values must be unique across all fabric maps. The **no fcmmap** command does not allow reversion to the default FCMAP value for a particular fabric map.

- (Optional) Change the advertisement (FCoE keep-alive, or FKA) interval from the default of 8000.

```
device(config-fcoe-fabric-map-SanA)# advertisement interval 10000
```

- Repeat the above for SanB, SanC, and SanD with appropriate values.

Use the **show fcoe fabric-map** command to confirm the current status of the FCoE fabric map, as in the following example.

```
device# show fcoe fabric-map SanA
=====
Fabric-Map VLAN VFID Pri FCMAP FKA Timeout
=====
SanA      1004 128[D] 4 0x0efc01 10000 Enabled[D]
Total number of Fabric Maps = 1
```

Configuring the FCF groups for FCoE logical SANs

The Access Gateways (AGs) must be configured to identify the membership of the FCoE Forwarder (FCF), as well as the RBridge IDs of the forwarding R Bridges. Those R Bridges must be first-hop switches. All intermediate switches forward the SAN traffic by default.

Do the following to configure the AGs to support FCoE logical SANs. Refer to [FCoE logical SAN use case 1](#) on page 25. Once the membership is configured, you must configure FCF groups and fabric maps for the nondefault SANs.

- In global configuration mode, enter the **fcoe** command to enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

- Enter the **fabric-map** command and specify a nondefault SAN, entering FCoE fabric-map configuration mode.

```
device(config-fcoe)# fabric-map SanA
device(config-fcoe-fabric-map-SanA)#
```

- In FCoE fabric-map configuration mode, specify an FCF group to enter FCoE FCF group configuration mode, and specify an FCF RBridge ID and one or more FIF RBridge IDs.

NOTE

Use the **add** keyword to add multiple RBridge IDs. Alternatively, use the **remove** keyword to remove RBridge IDs. Ranging (with a hyphen) is also allowed.

```
device(config-fcoe-fabric-map-SanA)# fcf-group rack-1
sw0(config-fabric-map-fcf-group-rack-1)# fcf-rbid 6
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid add 5
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid add 10-15
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid add 28-30,35
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid remove 28-30,12
```

4. You can verify the FCF map configuration by viewing the running configuration, or by entering the **show fcoe fcf-group** command as in the following examples:

```
device(config-fabric-map-fcf-group-rack-2)# do show running-config fcoe fabric-map fcf-group

fcoe
fabric-map sanA
  fcf-group rack-1
    fcf-rbid 100
    fif-rbid add 5,10-11,13-15,35
  !
  fcf-group rack-2
    fcf-rbid 150
    fif-rbid add 180
  !
!
fabric-map sanB
  fcf-group rack-3
    fcf-rbid 200
    fif-rbid add 220, 221

device(config-fabric-map-fcf-group-rack-2)# do show fcoe fcf-group
=====
FCF-Group      Fabric-Map  FCF_RBID      FIF_RBID(s)
=====
rack-1         sanA        100            5  10  11  13  14  15
               35
rack-2         sanA        150            180
rack-3         sanB        200            220, 221

Total number of FCF Groups = 3
```

Configuring FCoE logical SAN port groups

This task maps one or more local FCoE ports to a specified VLAN, enabling the addition or removal of FCoE Initialization Protocol (FIP) Forwarder (FIF) RBridge IDs to or from an FCoE Forwarder (FCF) group.

Currently only one local FCoE VLAN is supported. In a VCS Fabric cluster with multiple hosts and targets connected to switches by means of Flex ports, only the default FCoE VLAN (1002) can be used. Therefore, to map multiple local FC ports a particular (nondefault) VLAN, use the **fcport-group** and **fcport-group-rbid** commands. Note the following conditions:

- An FC port group is the same as an FCF group under a fabric map.
- An FC port group can only be created when the fabric map san-mode is "local".
- The FC port group has a 1:1 relationship with the fabric map.
- When the fabric map is deleted, the FC port group is deleted automatically.
- The **fcport-group-rbid** command, under the **fcport-group** command, is used to add and remove RBridge IDs.

Do the following to enable the addition or removal of FCoE Initialization Protocol (FIP) Forwarder (FIF) RBridge IDs to or from an FCoE Forwarder (FCF) group.

1. In global configuration mode, enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

2. Enter the fabric-map command and specify a logical SAN fabric map.

```
device(config-fcoe)# fabric-map SanA
device(config-fcoe-fabric-map-SanA)#
```

3. Enter the **fcport-group** command, to enter FCoE port-group configuration mode, and then enter the **fcport-group-rbid** command to specify one or more RBridge IDs.

```
device(config-fabric-map-fcport-group-SanA)# ?
Possible completions:
describe          Display transparent command information
do                Run an operational-mode command
exit              Exit from current mode
fcport-group-rbid Configure an FCPORT group rbridge-ids.
help              Provide help information
no                Negate a command or set its defaults
pwd               Display current mode path
top               Exit to top level and optionally run command
```

4. In FCoE port-group configuration mode, specify an RBridge ID.

```
device(config-fabric-map-fcport-group-SanA)# fcport-group-rbid add 29
```

NOTE

Ranging and comma delimiters are allowed for multiple RBridge ID entries. Use the **remove** keyword to remove one or more RBridge IDs.

5. To confirm the configuration, use the **show running-config-foe** command.

```
device(config-fabric-map-fcport-group-SanA)# do show running-config fcoe
fcoe
fabric-map default
  vlan 1002
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0E:FC:00
  advertisement interval 8000
  keep-alive timeout
!
fabric-map SanA
  vlan 4
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0E:FC:03
  advertisement interval 8000
  keep-alive timeout
  fcport-group
  fcport-group-rbid add 29
!
```

Assigning a fabric map onto an interface

The user assigns a fabric map onto an Ethernet interface by using the **fcoeport** command.

This configuration is called *FCoE provisioning*. Once the fabric map is assigned onto an interface, the following are applied to the interface:

- The corresponding FCoE VLAN
- The default CEE map
- The FCoE/FIP VLAN classifiers

In short, the interface becomes capable of carrying FCoE traffic.

NOTE

The fabric map can be applied irrespective of whether or not the interface is in "switchport" mode. However, the fabric map cannot be applied on an interface if the same interface already has a CEE map assigned to it.

Do the following to assign a fabric map onto an interface.

1. Activate interface subtype configuration mode for the interface to be modified.

```
device(config)# interface tengigabitethernet 1/0/1
device(conf-if-te-1/0/1)#
```

2. Apply the current fabric map to the interface by using the **fcoeport** command and entering "default" as the map name.

```
device(conf-if-te-1/0/1)# fcoeport default
```

NOTE

The default configuration continues to be supported. However, beginning with Network OS 6.0.0, user-specified map names are allowed to support FCoE logical SANs.

3. Return to privileged EXEC mode by using the **end** command.

```
device(conf-if-te-1/0/1)# end
```

4. Confirm the changes to the interface by using the **show running-config** command.

```
device# show running-config interface tengigabitethernet 1/0/1
interface TenGigabitEthernet 1/0/1
fcoeport default
no shutdown
```

5. Use the **fcoe fabric-map default** command to confirm the current status of the fabric map.

```
device# show fcoe fabric-map
=====
Fabric-Map VLAN      VFID  Pri  FCMAP      FKA      Timeout
=====
default      1002[D] 128[D] 3[D] 0xefc00[D] 8000[D] Enabled[D]
Total number of Fabric Maps = 1
```

6. Repeat this procedure for any additional interfaces as appropriate.

Assigning an FCoE fabric map onto a LAG member

The **fcoeport** command is used under interface subtype configuration mode to provision a port to be an FCoE port. This puts the port in Layer 2 mode, but only for FCoE VLANs. In Network OS 4.0.0 and later, the **fcoeport default** command is supported for LAG member ports where FCoE provisioning is applied to individual Ethernet ports.

For all LAGs with FSB, the **fcoeport config** command must be applied on the LAG itself. For all LAGs with directly attached CNAs, the **fcoeport config** command must be applied on the member ports. Once this command is applied, and if the member port of the LAG is CEE-capable, the port carries FCoE traffic only.

NOTE

Note the following conditions:

- FCoE provisioning is allowed on a LAG member only if the LAG is not FCoE provisioned.
- The default configuration continues to be supported. However, beginning with Network OS 6.0.0, user-specified fabric maps are allowed to support multiple FCoE logical SANs.

To assign the FCoE fabric map onto a LAG member, perform the following steps.

1. Activate interface subtype configuration mode for the interface to be modified.

```
device(config)# interface tengigabitethernet 3/0/19
device(conf-if-te-3/0/19)#
```

2. Activate the channel-group mode.

```
device(conf-if-te-3/0/19)# channel-group 10 mode active type standard
```

3. Set the LACP timeout to **long**.

```
device(conf-if-te-3/0/19)# lacp timeout long
```

4. Apply the current FCoE fabric map to the interface by using the **fcoeport default** command.

```
device(conf-if-te-3/0/19)# fcoeport default
```

5. Return to privileged EXEC mode by using the **end** command.

```
device(conf-if-te-3/0/19)# end
```

6. Confirm the changes to the interface with the **show running-config** command.

```
device# show running-config interface tengigabitethernet 3/0/19

interface TenGigabitEthernet 3/0/19
  fabric isl enable
  fabric trunk enable
  channel-group 10 mode active type standard
  lacp timeout long
  fcoeport default
  no shutdown
```

7. Use the **show fcoe interface brief** command to confirm the current status of the FCoE logins.

```
device# show fcoe interface brief
```

8. Repeat this procedure for additional interfaces as appropriate.

Configuring FCoE over LAG

Network OS 4.0.0 and later supports FCoE over LAGs. These are LAGs between the FCoE Forwarder (FCF) and a DCB-capable switch. The entire LAG is provisioned for FCoE, so that all member ports are used for FCoE traffic. FCoE traffic is broadcast on all the member links of the LAG.

NOTE

FCoE over LAG supports standard LAGs only; vLAGs are not supported.

Additionally, Network OS 4.0.0 and later supports multiple logins per port. This feature allows multiple ENodes to log in to a single 10-gigabit Ethernet port or a LAG.

Guidelines and restrictions for configuring FCoE over LAG

Follow these configuration guidelines and restrictions when configuring FCoE over LAG:

- The intermediate switches may or may not be an FSB. However, FSB is recommended for security.
- All ACLs and FCoE forwarding entries will continue to be on the FCF's ingress ports.

- It is assumed that the intermediate switch works in "Willing" mode towards the FCF in the DCBX exchange, and accepts the configuration from the FCF and propagates it downstream.
- The CEE/DCBX configuration is expected to be identical on both the FCF and the intermediate switch.
- Irrespective of the previous two items, the PFC/No-drop behavior from the FCF's perspective will be guaranteed only on the links between the FCF and the first-hop switch. There is no provision in the standard to guarantee this requirement on all paths leading to the Enode.
- FSBs may or may not be able to forward the FCoE LLS TLV to the Enodes. Hence this TLV may not be present in the LLDP packets sent to the Enodes. The FCF continues to send this TLV in its LLDP packets destined to the intermediate switch.
- The default map configuration continues to be supported on LAG port-channels, as shown in the following section. However, beginning with Network OS 6.0.0, user-specified fabric maps are allowed to support multiple FCoE logical SANs.

Configuring FCoE provisioning on LAGs

The **fcoeport** command has been extended to the LAG interfaces to support the logical SANs feature, as shown in the following example.

```
switch# configure
Entering configuration mode terminal

switch(config)# interface port-channel 10

switch(config-Port-channel-10)# fcoeport default

switch(config-Port-channel-10)#
```

This provisions all the member ports of port-channel 10 for FCoE.

NOTE

The default configuration continues to be supported. (The keyword "default" must be entered manually.) However, beginning with Network OS 6.0.0, user-specified fabric maps are allowed to support multiple FCoE logical SANs.

Configuring interfaces to support FCoE logical SANs

To assign the logical FCoE fabric maps onto an interface, use the logical SAN fabric map instead of the "default" map as the map name in the **fcoeport** command. The guidelines for provisioning FCoE on physical interfaces, LAG members, and LAGs is the same as for the default case.

The following example provisions the 10-gigabit Ethernet interface in slot 0, port 1 on RBridge ID 1.

1. Activate interface configuration mode for the interface to be modified.

The following example activates the mode for the 10-gigabit Ethernet interface in slot 0/port 1 on RBridge ID 1.

```
switch(config)# interface tengigabitethernet 1/0/1
switch(conf-if-te-1/0/1)#
```

2. Apply the current FCoE fabric map to the interface by using the **fcoeport** command with the desired map name.

```
switch(conf-if-te-1/0/1)# fcoeport SanA
```

3. Return to privileged EXEC mode by using the **end** command.

```
switch(conf-if-te-1/0/1)# end
switch#
```

4. Confirm the changes to the interface by using the **show running-config** command.

```
switch# show running-config interface tengigabitethernet 1/0/1
interface TenGigabitEthernet 1/0/1
 fcoeport SanA
 no shutdown
```

5. Repeat this procedure for additional interfaces as appropriate.

Verifying FCoE logical SAN configurations

The following table lists **show** commands that can be used to verify FCoE logical SAN configurations. For details, refer to the *Extreme Network OS Command Reference*.

TABLE 3 Commands to verify FCoE logical SAN configurations

Command	Description
show fcoe devices	Displays the FCoE devices information.
show fcoe fabric-map	Displays the FCoE fabric-map configuration globally in a fabric, or on a single RBridge.
show fcoe fcf-group	Displays FCF groups information.
show fcoe fcoe-enodes	Displays FCoE ENodes information.
show fcoe fcoe-map	Displays information about the FCoE map.
show fcoe interface ethernet	Displays a synopsis of the FCoE Ethernet interfaces.
show fcoe fcpport-group	Displays RBridge IDs and FCoE port connector groups associated with a fabric map.
show fcoe login	Displays FCoE login information.

FCoE logical SANs configuration examples

Use case 2 configuration example

The VLANs are assigned as follows.

VLAN	Fabric	Logical SAN
1002	Default	Local
1003	fabric map sanA	Remote
1004	fabric map sanB	Remote
1005	fabric map sanC	Local
1006	fabric map sanD	Local

```
fcoe
fabric-map default
vlan 1002
san-mode local
priority 3
virtual-fabric 128
fcmmap 0E:FC:00
advertisement interval 8000
keep-alive timeout
!

fabric-map sanA
vlan 1003
san-mode remote
priority 3
virtual-fabric 128
fcmmap 0e:fc:10
advertisement interval 8000
keep-alive timeout
fcf-group sanA
fcf-rbid 1
fif-rbid add 2
!

fabric-map sanB
vlan 1004
san-mode remote
priority 3
virtual-fabric 128
fcmmap 0e:fc:20
advertisement interval 8000
keep-alive timeout
fcf-group sanB
fcf-rbid 3
fif-rbid add 4
!

fabric-map sanC
vlan 1005
san-mode local
priority 3
virtual-fabric 128
fcmmap 0e:fc:30
advertisement interval 8000
keep-alive timeout
!

fabric-map sanD
vlan 1006
```

```
san-mode local
priority 3
virtual-fabric 128
fcmap 0e:fc:40
advertisement interval 8000
keep-alive timeout
```

Use case 3 configuration example

The VLANs are assigned as follows.

VLAN	Fabric	Logical SAN
1002	Default	Local
1003	fabric map sanA	Local
1004	fabric map sanB	Local
1005	fabric map sanC	Local
1006	fabric map sanD	Local

```

fcoe
fabric-map default
vlan 1002
san-mode local
priority 3
virtual-fabric 128
fcmmap 0E:FC:00
advertisement interval 8000
keep-alive timeout
!

fabric-map sanA
vlan 1003
san-mode local
priority 3
virtual-fabric 128
fcmmap 0e:fc:10
advertisement interval 8000
keep-alive timeout
!

fabric-map sanB
vlan 1004
san-mode local
priority 3
virtual-fabric 128
fcmmap 0e:fc:20
advertisement interval 8000
keep-alive timeout
!

fabric-map sanC
vlan 1005
san-mode local
priority 3
virtual-fabric 128
fcmmap 0e:fc:30
advertisement interval 8000
keep-alive timeout
!

fabric-map sanD
vlan 1006
san-mode local
priority 3
virtual-fabric 128
fcmmap 0e:fc:40
advertisement interval 8000
keep-alive timeout
!

```

Managing duplicate WWNs

This feature enables the management of conflicts resulting from duplicate port WWNs arising from multiple causes.

It is never appropriate for two devices with the same port WWN (WWPN) to be in the same Name Server at the same time, as this results in incorrect responses to commands and unpredictable results, for a variety of reasons:

- *Timing issues:* Because Fibre Channel data bases are distributed, one device can appear to be logged in through two unique port locations.
- *Notification errors:* The removal of a device is not recognized and the device logs back in with a different port address, as if there were two devices.
- *Actual duplication:* Two distinct devices with the same WWPN attempt to log into the fabric at the same time. (This issue is not resolvable without authenticating the devices or using access control to limit which devices are allowed to log into specific ports.)

Beginning with Network OS 6.0.1, the user can specify system behavior when duplicate WWPNs are detected, by using the **fabric login-policy** command on a specified RBridge. This is a node-specific cluster command whose effect is persistent across reboots. A secondary node must be configured only through the principal node in management cluster or logical chassis cluster modes. When duplicate WWPNs are detected across two switches, the Name Server posts RASLOG NS-1012.

NOTE

This feature is not available on a device that is configured for Access Gateway mode.

The keywords below enable the following options:

Keyword	Description
new-login	Allows the "new" device to log in and cleans up the "old" (previous) login.
old-login	Allows the "old" device to retain the login and rejects the "new" login. This is the default.

NOTE

For reasons noted above, the concepts "old" and "new" are relative and not necessarily literal. The methods required to correct actual duplicate WWPN conditions within a fabric are beyond the scope of this feature.

To view the current configuration of this feature, use the **show fabric login-policy** command to view the login policy for a local node, a specific RBridge, or all RBridges in the fabric.

802.1Q VLANs

• 802.1Q VLAN overview.....	49
• Configuring and managing 802.1Q VLANs.....	51
• Private VLANs.....	60
• Protected port for uplinks	63

802.1Q VLAN overview

NOTE

This chapter addresses the use of standard Virtual LANs (VLANs) as defined by IEEE 802.1Q. Those VLANs have VLAN IDs that range from 1 through 4096.

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.

NOTE

The VLAN used for carrying FCoE traffic needs to be explicitly designated as the FCoE VLAN. FCoE VLANs are configured through the Network OS CLI (refer to the Configuring an interface port as a Layer 2 switch port section for details). Currently only one VLAN can be configured as the FCoE VLAN at a time.

Ingress VLAN filtering

A frame arriving at Extreme VDX hardware is either associated with a specific port or with a VLAN, based on whether the frame is tagged or untagged. The association rules are as follows:

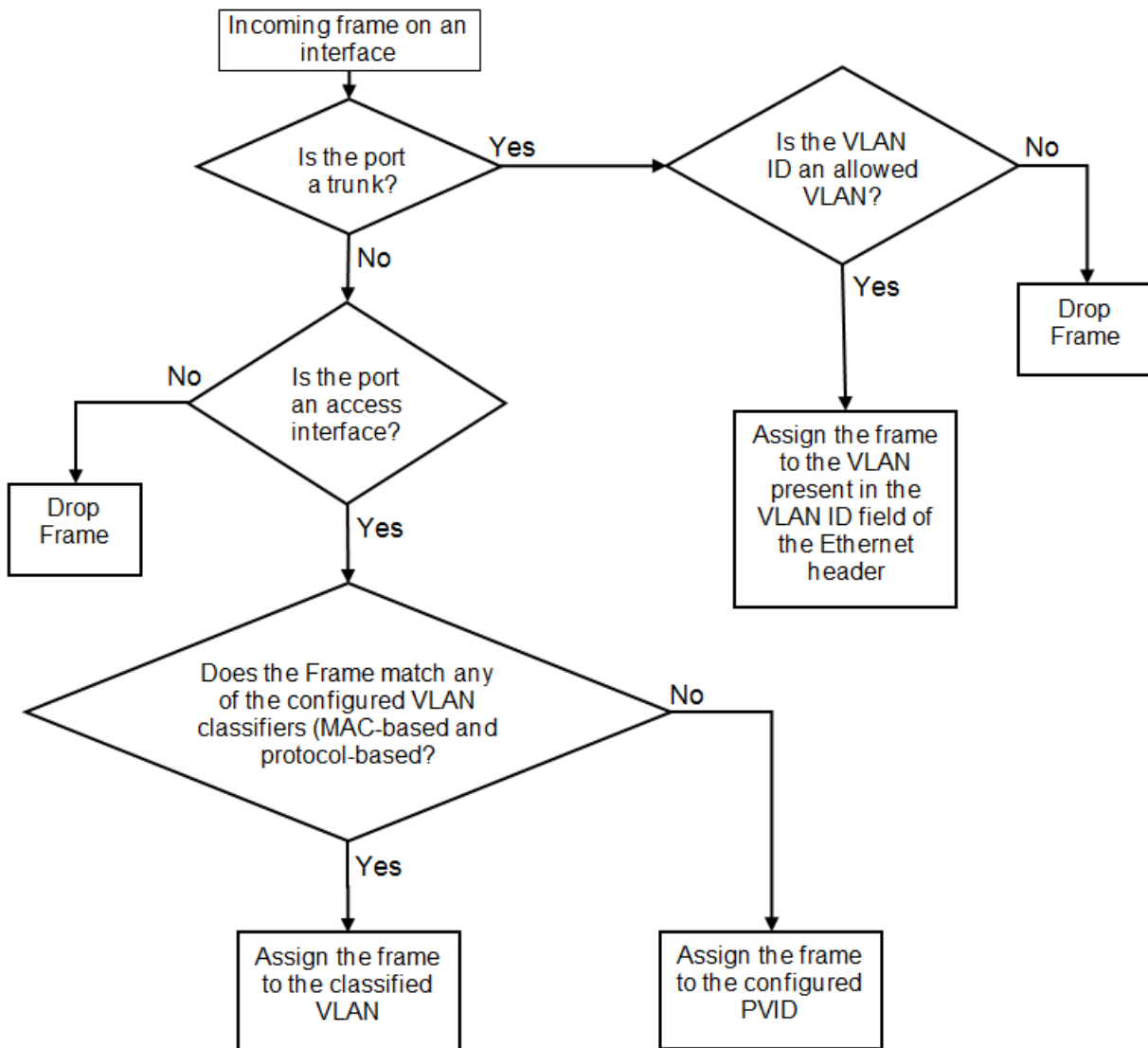
- Admit tagged frames only — The port the frame came in on is assigned to a single VLAN or to multiple VLANs depending on the VLAN ID in the frame's VLAN tag. This is called trunk mode.
- Admit untagged frames only — These frames are assigned the port VLAN ID (PVID) assigned to the port the frame came in on. This is called access mode.
- Admit VLAN tagged and untagged frames — All tagged and untagged frames are processed as follows:
 - All untagged frames are classified into native VLANs.
 - If the tenGigabitEthernet interface port is configured as an fcoeport and is in access mode, untagged Layer 2 or priority-tagged frames are forwarded by the egress port as untagged frames, unless you enable priority-tagging on the tenGigabitEthernet interface. By default, priority-tagging is disabled.
 - Any tagged frames coming with a VLAN tag equal to the configured native VLAN are processed.
 - For ingress and egress, non-native VLAN tagged frames are processed according to the allowed VLAN user specifications. This is called trunk mode.

NOTE

Ingress VLAN filtering is enabled by default on all Layer 2 interfaces. This ensures that VLANs are filtered on the incoming port (depending on the user configuration).

The following illustrates the frame-processing logic for an incoming frame.

FIGURE 6 Ingress VLAN filtering



There are important facts you should know about Ingress VLAN filtering:

- Ingress VLAN filtering is based on port VLAN membership.
- Port VLAN membership is configured through the Network OS CLI.
- Dynamic VLAN registration is not supported.
- The Extreme VDX hardware does VLAN filtering at both the ingress and egress ports.
- The VLAN filtering behavior on logical Layer 2 interfaces such as LAG interfaces is the same as on port interfaces.
- The VLAN filtering database (FDB) determines the forwarding of an incoming frame.

Additionally, there are important facts you should know about the VLAN FDB:

- The VLAN FDB contains information that helps determine the forwarding of an arriving frame based on MAC address and VLAN ID data. The FDB contains both statically configured data and dynamic data that is learned by the switch.
- The dynamic updating of FDB entries using learning is supported (if the port state permits).
- Dynamic FDB entries are not created for multicast group addresses.
- Dynamic FDB entries are aged out based on the aging time configured per Extreme VDX hardware. The aging time is between 60 and 1000000 seconds. The default is 300 seconds.
- You can add static MAC address entries specifying a VLAN ID. Static entries are not aged out.
- A static FDB entry overwrites an existing dynamically learned FDB entry and disables learning of the entry going forward.

NOTE

For more information on frame handling for Extreme VDX hardware, refer to [FCoE and Layer 2 Ethernet](#) on page 16.

VLAN configuration guidelines and restrictions

Follow these guidelines and restrictions when configuring VLANs:

- On all Extreme VDX switches, VLAN 1002 is reserved for FCoE VLAN functionality.
- On Extreme VDX 8770 switches: 1 through 4086 for 802.1Q VLANs (VLAN IDs 4087 through 4095 are reserved on these switches), and 4096 through 8191 for service or transport VFs in a Virtual Fabrics context.
- On all other Extreme VDX switches: 1 through 3962 for 802.1Q VLANs (VLAN IDs 3963 through 4095 are reserved on these switches), and 4096 through 8191 for service or transport VFs in a Virtual Fabrics context.
- In an active topology, MAC addresses can be learned, per VLAN, using Independent VLAN Learning (IVL) only.
- A MAC address ACL always overrides a static MAC address entry. In this case, the MAC address is the forwarding address and the forwarding entry can be overwritten by the ACL.
- The Extreme DCB switch supports Ethernet DIX frames and 802.2 LLC SNAP encapsulated frames only.
- You must configure the same native VLAN on both ends of an 802.1q trunk link. Failure to do so can cause bridging loops and VLAN leaks.
- All switches in a cluster must be configured with the same VLAN number.

Configuring and managing 802.1Q VLANs

Understanding the default VLAN configuration

The following table summarizes the default VLAN configuration. Consider this when making configuration changes.

TABLE 4 Default VLAN configuration

Parameter	Default setting
Default VLAN	VLAN 1
Interface VLAN assignment	All interfaces assigned to VLAN 1
VLAN state	Active
MTU size	2500 bytes

Configuring interfaces to support VLANs

This section details the various tasks required to configure and manage VLAN traffic.

Enabling and disabling an interface port

NOTE

DCB interfaces are enabled by default in Extreme VCS Fabric mode.

NOTE

DCB interfaces do not support auto-negotiation of Ethernet link speeds. The DCB interfaces support 10-gigabit Ethernet and 1-gigabit Ethernet.

To enable and disable an interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

Configuring the MTU on an interface port

NOTE

The entire fabric acts like a single switch. Therefore, MTU is applicable only on the edge-ports, and not on ISL.

To configure the maximum transmission unit (MTU) on an interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **mtu** command to specify the MTU value on the interface port.

```
switch(conf-if-te-0/1)# mtu 4200
```

Creating a VLAN

On Extreme VDX hardware, VLANs are treated as interfaces from a configuration point of view.

By default all the DCB ports are assigned to VLAN 1 (VLAN ID equals 1). The *vlan_ID* value can be 1 through 4095. Refer to [VLAN configuration guidelines and restrictions](#) on page 51 for additional information.

To create a VLAN interface, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface vlan** command to assign the VLAN interface number.

```
switch(config)# interface vlan 1010
```

Enabling STP on a VLAN

Once all of the interface ports have been configured for a VLAN, you can enable Spanning Tree Protocol (STP) for all members of the VLAN with a single command. Whichever protocol is currently selected is used by the VLAN. Only one type of STP can be active at a time.

A physical interface port can be a member of multiple VLANs. For example, a physical port can be a member of VLAN 1015 and VLAN 55 simultaneously. In addition, VLAN 1015 can have STP enabled and VLAN 55 can have STP disabled simultaneously.

To enable STP for a VLAN, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **protocol spanning tree** command to select the type of STP for the VLAN.

```
switch(config)# protocol spanning tree mstp
```

3. Enter the **interface** command to select the VLAN interface number.

```
switch(config)# interface vlan 1015
```

4. Enter the **no spanning-tree shutdown** command to enable spanning tree on VLAN 1015.

```
switch(conf-if-vl-1015)# no spanning-tree shutdown
```

Disabling STP on a VLAN

Once all of the interface ports have been configured for a VLAN, you can disable STP for all members of the VLAN with a single command.

To disable STP for a VLAN, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to select the VLAN interface number.

```
switch(config)# interface vlan 55
```

3. Enter the **spanning-tree shutdown** command to disable STP on VLAN 55.

```
switch(conf-if-vl-55)# spanning-tree shutdown
```

Configuring an interface port as a Layer 2 switch port

To configure the interface as a Layer 2 switch port, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **switchport** command to configure the interface as a Layer 2 switch port.

```
switch(config-if-te-1/0/1)# switchport
```

4. Enter the **do show** command to confirm the status of the DCB interface.

```
switch(conf-if-te-1/0/1)# do show interface tengigabitethernet 1/0/1
```

5. Enter the **do show** command to confirm the status of the DCB interface running configuration.

```
switch(conf-if-te-1/0/1)# do show running-config interface tengigabitethernet 1/0/1
```

Configuring an interface port as an access interface

Each DCB interface port supports admission policies based on whether the frames are untagged or tagged. Access mode admits only untagged and priority-tagged frames.

To configure the interface as an access interface, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **switchport** command to make the interface a Layer 2 switch port.

```
switch(conf-if-te-0/1)# switchport
```

4. Configure the interface as an access interface.

```
switch(conf-if-te-0/1)# switchport mode access
```

5. Enter the **switchport** command again to configure the DCB interface as a VLAN.

```
switch(conf-if-te-0/1)# switchport access vlan 20
```

Configuring an interface port as a trunk interface

Each DCB interface port supports admission policies based on whether the frames are untagged or tagged. Trunk mode admits only VLAN-tagged frames.

To configure the interface as a trunk interface, run the following steps in privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/19
```

3. Enter the **switchport** command to place the DCB interface into trunk mode.

```
switch(conf-if-te-1/0/19)# switchport mode trunk
```

4. Specify whether all, one, or none of the VLAN interfaces are allowed to transmit and receive through the DCB interface. Enter the one of the following commands as is appropriate for your needs.

Allowing only one VLAN to transmit or receive through the DCB interface

This example allows VLAN 30 to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan add 30
```

Allowing all VLANs to transmit or receive through the DCB interface

This example allows all VLANs to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan all
```

Excluding a VLAN from the DCB interface

This example allows all except VLAN 11 to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan except 11
```

Blocking all VLANs from the DCB interface

This example allows none of the VLANs to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan none
```

Disabling a VLAN on a trunk interface

To disable a VLAN on a trunk interface, perform the following steps from privileged EXEC mode.

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

```
switch(config)# interface tengigabitethernet 1/0/10
```

3. Enter the **switchport** command to place the DCB interface into trunk mode.

```
switch(conf-if-te-1/0/10)# switchport mode trunk
```

4. Enter the **switchport** command again to remove the VLAN ranges from the trunk port.

```
switch(conf-if-te-1/0/10)# switchport trunk allowed vlan remove 30
```

Configuring protocol-based VLAN classifier rules

You can configure VLAN classifier rules to define specific rules for classifying frames to selected VLANs based on protocol and MAC addresses. Sets of rules can be grouped into VLAN classifier groups (refer to [Deleting a VLAN classifier rule](#) on page 56).

VLAN classifier rules (1 through 256) are a set of configurable rules that reside in one of these categories:

- 802.1Q protocol-based classifier rules
- Source MAC address-based classifier rules
- Encapsulated Ethernet classifier rules

NOTE

Multiple VLAN classifier rules can be applied per interface, provided that the resulting VLAN IDs are unique for the different rules.

802.1Q protocol-based VLANs apply only to untagged frames, or frames with priority tagging.

With both Ethernet-II and 802.2 SNAP encapsulated frames, the following protocol types are supported:

- Ethernet hexadecimal (0x0000 through 0xffff)
- Address Resolution Protocol (ARP)

- Fibre Channel over Ethernet (FCoE)
- FCoE Initialization Protocol (FIP)
- IP version 4 (IPv4)
- IP version 6 (IPv6)

NOTE

For complete information on all available VLAN classifier rule options, refer to the *Extreme Network OS Command Reference*.

Configuring a VLAN classifier rule

To configure a ARP protocol-based VLAN classifier rule, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **vlan classifier rule** command to configure a protocol-based VLAN classifier rule.

```
switch(config)# vlan classifier rule 1 proto ARP encap ethv2
```

NOTE

Refer to the *Extreme Network OS Command Reference* for complete information on all the protocols available for the **vlan classifier rule** command.

Configuring MAC address-based VLAN classifier rules

To configure a MAC address-based VLAN classifier rule, perform the following steps from privileged EXEC mode.

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

```
switch# configure terminal
```

2. Enter the **vlan classifier rule** command to configure a MAC address-based VLAN classifier rule.

```
switch(config)# vlan classifier rule 5 mac 0008.744c.7fid
```

Deleting a VLAN classifier rule

VLAN classifier groups (1 through 16) can contain any number of VLAN classifier rules.

To configure a VLAN classifier group and remove a VLAN classifier rule, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Specify a VLAN classifier group and delete a rule.

```
switch(config)# vlan classifier group 1 delete rule 1
```

Creating a VLAN classifier group and adding rules

VLAN classifier groups (1 through 16) can contain any number of VLAN classifier rules.

To configure a VLAN classifier group and add a VLAN classifier rule, perform the following steps from privileged EXEC mode.

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

2. Create a VLAN classifier group and add a rule.

```
switch(config)# vlan classifier group 1 add rule 1
```

Activating a VLAN classifier group with an interface port

To associate a VLAN classifier group with an interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and RBridge/slot/port number.

```
switch(config)# interface tengigabitethernet 22/0/10
```

3. Enter the **vlan classifier** command to activate and associate it with a VLAN interface (group 1 and VLAN 2 are used in this example).

```
switch(conf-if-te-22/0/10)# vlan classifier activate group 1 vlan 2
```

NOTE

This example assumes that VLAN 2 was already created.

Displaying VLAN information

NOTE

The **show vlan brief** command displays the VLAN as inactive if there are no member ports associated to that VLAN, or if the ports associated are in an admin down state.

To display VLAN information, perform one or both the following steps from privileged EXEC mode.

1. To display the configuration and status of the specified interface, enter the **show interface** command.

```
switch# show interface tengigabitethernet 3/0/10
```

2. To display the specified VLAN information, enter the **show vlan** command.

For example, this syntax displays the status of VLAN 20 for all interfaces, including static and dynamic:

```
switch# show vlan 20
```

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
```

Disabling source MAC learning on an interface

You can disable legacy dynamic MAC address learning on one or more VLANs on an interface. (CML is disabled by default.) VLANs range from 1 through 4090 for 802.1Q VLANs, and from 4096 through 8191 for VLANs in a Virtual Fabrics context.

To add a VLAN and range of VLANs to a switchport trunk:

```
switch(conf-if-te-4/0/5)# switchport trunk allowed vlan add 2000,3000-3500
```

To add the above VLANs to the MAC-learning-disabled list:

```
switch(conf-if-te-4/0/5)# mac-learning disable vlan add 2000,3000-3500
```

To remove a VLAN from the list:

```
switch(conf-if-te-4/0/5)# mac-learning disable vlan remove 2000
```

To disable MAC learning for all VLANs on the interface:

```
switch(conf-if-te-4/0/5)# no mac-learning disable
```

To disable MAC learning for VLANs 2000, 3000, and 3001 on the interface:

```
switch(conf-if-te-4/0/5)# mac-learning disable vlan 2000, 3000, 3001
```

To view the status of the VLANs, use the **show interface switchport** command, as in the following example:

```
switch# show interface tengigabitethernet 4/0/5 switchport

Interface name           : TenGigabitEthernet 4/0/5
Switchport mode          : trunk
Fcoeport enabled         : no
Ingress filter           : enable
Acceptable frame types   : vlan-tagged only
Native Vlan              : 1
Active Vlans             : 1-2,4,7-10
Inactive Vlans           : -
MAC learn disable Vlans  : 2000,3000,3001
```

Private VLANs

A private VLAN (PVLAN) domain is built with at least one pair of VLAN IDs; one (and only one) primary VLAN ID plus one or more secondary VLAN IDs. A primary VLAN is the unique and common VLAN identifier of the whole private VLAN domain and of all its VLAN ID pairs. Secondary VLANs can be configured as one of two types: either isolated VLANs or community VLANs. Up to 24 isolated or community VLANs can be part of a PVLAN domain.

An isolated VLAN is a secondary VLAN whose distinctive characteristic is that all hosts connected to its ports are isolated at Layer 2. A community VLAN is a secondary VLAN that is associated to a group of ports that connect to a designated community of end devices with mutual trust relationships.

A PVLAN is often used to isolate networks from security attacks, or to simplify IP address assignments.

Within the private VLAN, ports can be assigned port types. A port can be assigned to only one kind of port type at a time. The types of ports available for private VLANs are described in the following table.

TABLE 5 Private VLAN terms and definitions

Term	Description
Isolated port	An isolated port cannot talk to any other port in the private VLAN domain except for promiscuous ports and traffic ports. If a customer device needs to have access only to a gateway router, then it should be attached to an isolated port.
Community port	A community port is part of a group of ports that have Layer 2 communications with one another, and can also talk to any promiscuous port. For example, if you have two devices that you want to be isolated from other devices, but still be able to communicate between themselves, then community ports should be used. You cannot configure multiple community VLANs on a single port.
Promiscuous port	A promiscuous port can talk to all other types of ports. A promiscuous port can talk to isolated ports as well as community ports. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with the customer endpoints are typically connected using promiscuous ports.
Trunk port	A trunk port connects two switches and carries two or more VLANs.
Promiscuous trunk port	A promiscuous trunk port carries multiple primary and normal VLANs. Packets are received and transmitted with primary or regular VLAN tags. Otherwise, the port operates as a promiscuous port.
Secondary VLAN	A VLAN used to implement PVLANs. Secondary VLANs are associated with a primary VLAN, and carry traffic from hosts to other allowed hosts or routers.
Community VLAN	A secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways, and to other host ports in the same community. Multiple community VLANs are permitted in a PVLAN.
Primary VLAN	A PVLAN has only one primary VLAN. Every port in a PVLAN is a member of the primary VLAN. The primary VLAN carries unidirectional traffic downstream from the promiscuous ports to the isolated and community ports and to other promiscuous ports.

PVLAN configuration guidelines and restrictions

Follow these guidelines and restrictions when configuring VLANs:

- VE configuration is not supported on a primary VLAN.
- IGMP is not supported on private VLANs; however you can create an IGMP configuration. The configuration succeeds but the hardware is not programmed.
- For private VLANs, egress ACLs on the primary VLAN are applied only for the traffic that ingresses and egresses from the primary VLAN, and not for the traffic that gets translated from the secondary VLAN to the primary VLAN.
- For private VLANs, egress ACLs on the primary VLAN are also applied to the traffic that gets translated to the secondary VLAN.
- STP is not supported on private VLAN host ports.

Associating the primary and secondary VLANs

This procedure configures the PVLAN and associates the secondary VLAN with the primary VLAN.

1. Configure the VLAN interface.

```
switch(config)# interface vlan 10
```

2. Configure the VLAN as a primary PVLAN.

```
switch(conf-if-vl-10)# private-vlan primary
```

3. Create multiple community VLANs, for use in Step 5.

```
switch(config)# interface vlan 100
switch(conf-if-vl-100)# private-vlan community
```

4. Configure the secondary VLAN (isolated).

```
switch(config)# interface vlan 200
switch(conf-if-vl-200)# private-vlan isolated
```

5. Associate the multiple community VLANs and the isolated VLAN.

```
switch(config)# interface vlan 10
switch(conf-if-vl-10)# private-vlan association add 100,200
```

6. Exit VLAN configuration mode.

```
switch(conf-if-vl-10)# exit
```

Configuring an interface as a PVLAN promiscuous port

This procedure configures an interface as the PVLAN promiscuous port.

1. Specify the interface.

```
switch(config)# interface tengigabitethernet 0/1
```

2. Mark the interface as switch port

```
switch(conf-if-te-0/1)# switchport
```

3. Configure the interface as a PVLAN promiscuous port (untagged).

```
switch(conf-if-te-0/1)# switchport mode private-vlan promiscuous
```

4. Configure the interface as a PVLAN promiscuous port (tagged).

```
switch(conf-if-te-0/1)# switchport mode private-vlan trunk promiscuous
```

5. Associate the interface with a PVLAN.

```
switch(conf-if-te-0/1)# switchport private-vlan mapping 10 add 100,200
```

6. Configure a normal VLAN on the PVLAN promiscuous port.

```
switch(conf-if-te-0/1)# switchport trunk allowed vlan add 500
```

Configuring an interface as a PVLAN host port

This procedure configures an interface as the PVLAN host port and thereby isolates a VLAN.

1. Specify the interface.

```
switch(config)#interface tengigabitethernet 1/0/1
```

2. Mark the interface as a switch port.

```
switch(conf-if-te-1/0/1)# switchport
```

3. Configure the interface as a PVLAN host port that is tagged.

```
switch(conf-if-te-1/0/1)# switchport mode private-vlan trunk host
```

4. Alternatively, configure the interface as a PVLAN host port that is untagged.

```
switch(conf-if-te-1/0/1)# switchport mode private-vlan host
```

5. Associate the interface with a PVLAN and isolate VLAN 100.

```
switch(conf-if-te-1/0/1)# switchport private-vlan host-association 10 100
```

Configuring an interface as a PVLAN trunk port

This procedure configures an interface as a PVLAN trunk port.

1. Specify the interface.

```
switch(config)# interface tengigabitethernet 0/1
```

2. Mark the interface as switch port.

```
switch(conf-if-te-0/1)# switchport
```

3. Configure the interface as a PVLAN trunk port.

Do not complete this step if the host is a plain, untagged server.

```
switch(conf-if-te-0/1)# switchport mode private-vlan trunk
```

4. Configure the association between primary VLANs and secondary VLANs and the PVLAN trunk port with a PVLAN.

```
switch(conf-if-te-0/1)# switchport private-vlan association trunk 10 100
```

NOTE

Multiple PVLAN pairs can be specified by means of the **switchport private-vlan association trunk** command, so that a PVLAN trunk port can carry multiple secondary VLANs. If an association is specified for the existing primary VLAN, the existing association is replaced. If there is no trunk association, any packets received on secondary VLANs are dropped.

5. Configure a normal VLAN on the PVLAN trunk port.

```
switch(conf-if-te-0/1)# switchport private-vlan trunk allowed vlan 400
```

- Configure a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port. If there is no native VLAN configured, all untagged packets are dropped. If the native VLAN is a secondary VLAN and the port does not have the association for the secondary VLAN, the untagged packets are dropped.

```
switch(conf-if-te-0/1)# switchport private-vlan trunk native vlan 600
```

Displaying PVLAN information

To display private VLAN information, use the **show vlan private-vlan** command to see the private VLAN types (for example, "private," "isolated," and "community," as in the following example).

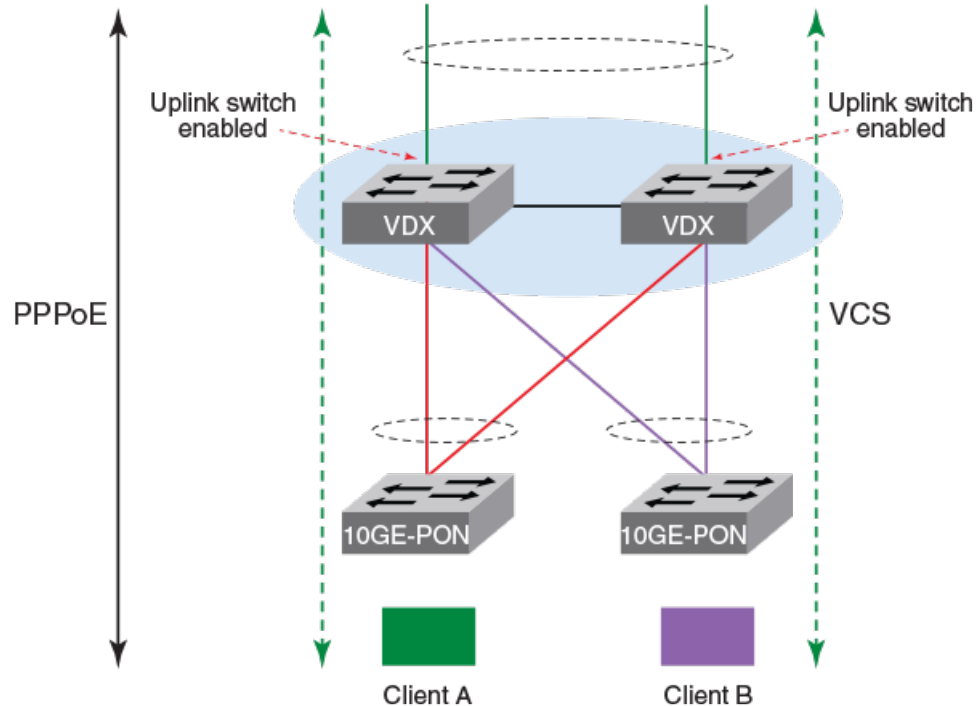
```
device# show vlan private-vlan
```

Protected port for uplinks

Protected port for uplinks provides Layer 2 isolation between interfaces (Ethernet ports and LAGs) that share the same broadcast domain, or VLAN.

The following figure illustrates Layer 2 traffic flow (PPPoE) between multiple clients and an uplink port. This isolated behavior is typical in the case of a PVLAN domain, where two or more isolated ports cannot communicate with each other, and ensures that the only possible communication is through a promiscuous port.

FIGURE 7 Layer 2 traffic between multiple clients and an uplink port



The protected port feature could apply, for example, to students in dormitories or where there are multiple dwelling units that share access. Broadcast storm damage is reduced by limiting broadcast storm traffic to a single switchport and uplink, facilitating traffic diagnosis.

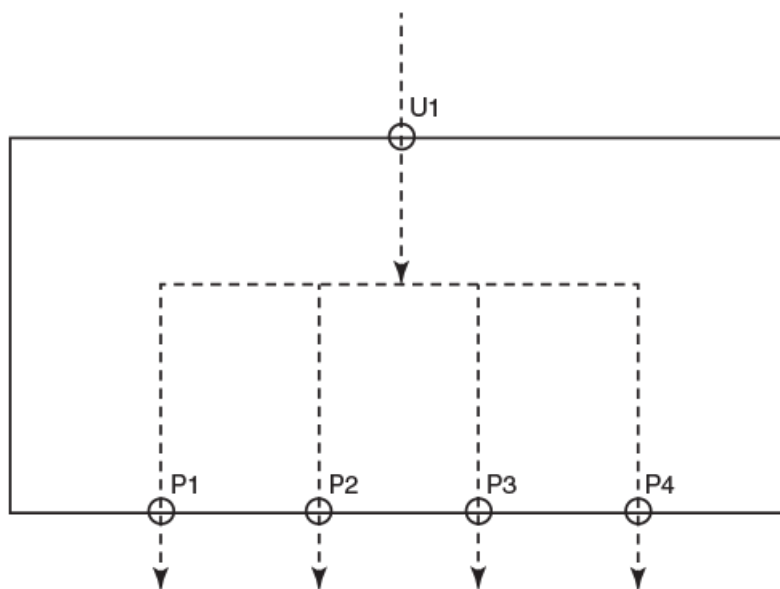
Physical ports, LAGs, and vLAGs can be configured as protected ports as well as uplink ports. Multiple uplink ports are supported.

Packets received from protected ports can be forwarded only to unprotected egress ports. Protected port filtering rules are also applied to packets that are forwarded by software, such as snooping applications. Port protection is not subject to VLAN membership. Devices connected to protected ports are not allowed to communicate with each other, even if they are members of the same VLAN. Both physical ports and LAGs can be defined as protected or unprotected.

VLAN Layer 2 forwarding

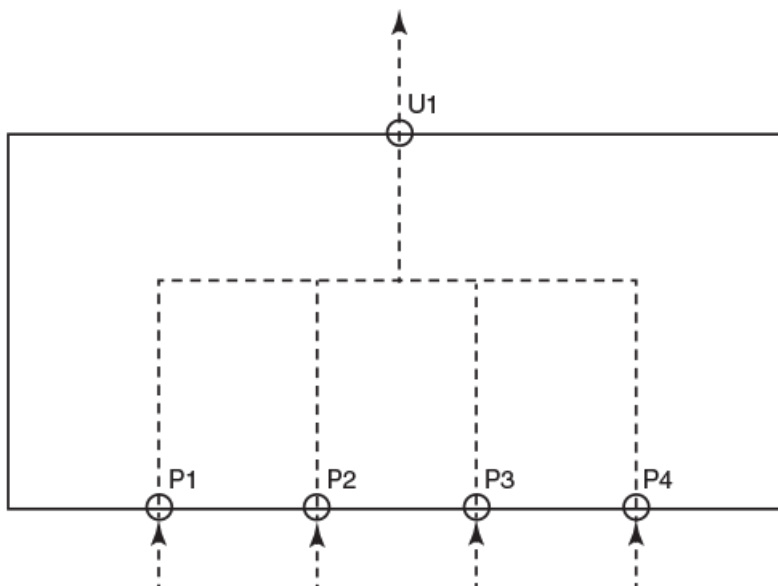
For an incoming packet, uplink and protected ports accept only VLAN tag, priority tag, or untagged packets. Protected ports cannot talk to each other in the Layer 2 domain. The following figures illustrate the behavior of uplink and protected ports, respectively.

FIGURE 8 Uplink port behavior



Uplink port U1 can send packets to all protected ports P1, P2, P3, and P4. Uplink ports and all protected ports belong to the same VLAN.

FIGURE 9 Protected port behavior



All protected ports P1, P2, P3, and P4 can send packets to uplink port U1. However, traffic is not allowed among the protected ports. The uplink and all protected ports belong to same VLAN.

Protected port behavior

Note the following behavior for protected ports:

- To enable this feature on an interface, you must enable protected port for uplinks globally on the switch by using the global **uplink-switch enable** command.
- When a port is made protected (by means of the **protected-port enable** command), the port is removed from the VLAN to which it belongs and is assigned to an internal VLAN in the range 7168 through 8191. The same port is added as part of this new internal VLAN. For example, if interface Te 1/0/1 is part of VLAN 10 and the **protected-port enable** command is used on the interface, Te 1/0/1 is removed from VLAN 10 and is added to internal VLAN 7168. Up to 1024 VLANs are supported.
- The new internal VLAN is just an isolated VLAN in the case of a PVLAN and behaves the same way as an isolated VLAN, thus preventing communication among protected ports.
- The internal VLANs and their membership (as protected ports) are shown in the output of the **show vlan brief** command.
- Both dot1q and global VLANs (GVLANS) can be configured on a protected port.

Limitations and considerations

Note the following additional considerations and limitations:

- Virtual Fabrics (VF) must be enabled on the switch to enable the global **uplink-switch enable** command.
- VLANs 7168 through 8191 are reserved internally when the global command is used, and these VLANs are not allowed to be created by the user.
- The VLAN/VF configured should be same on both protected and uplink ports. Enabling protected port is not allowed if VLANs are not configured on the interface.
- By default, all switchports are in unprotected mode, which is just an uplink port mode.

- At least one uplink port must be present to enable a protected port configuration. This means that two ports with same VLAN must be configured for the **protected-port enable** command to apply. In case of a vLAG, each node of the vLAG must have at least one uplink port in order to have a successful protected-port configuration on the vLAG.
- In the case of VCS, one uplink port must be present for each RBridge.
- In the case of VCS with multiple nodes, if protected ports with a specific VLAN are required on all RBridges, the user must configure at least one uplink port and one protected port for that specific VLAN on all RBridges before proceeding to configure a new VLAN on the protected ports. The same mapping of the user VLAN to the internal VLAN on all RBridges is mandatory because the selection of an internal VLAN for each user-configured VLAN is local to each RBridge. The same configuration order must be followed when removing a specific VLAN from one or more protected ports.
- The above two points are applicable to a VCS with multiple nodes, as well as if uplink and protected ports are required in all or multiple nodes. However, this is not applicable if all the uplink and protected ports belong to a same RBridge.
- All varieties of STP must be blocked on a VLAN that contains one or more protected ports.
- The following configurations are not supported on a VLAN that contains one or more protected ports: Layer 3 routing, ARP, IGMP, IP configuration, VE configuration, AMPP, PVLAN, RSPAN, and ACLs.

Configuring protected port for uplinks

This task shows the global and interface configurations to enable protected port.

1. Enter global configuration mode and enter the **uplink-switch enable** command.

```
device# configure terminal
device(config)# uplink-switch enable
```

2. Specify an Ethernet interface and enter the **switchport** command to specify the interface as a switchport.

```
device(config)# interface tengigabitethernet 1/0/2
device(conf-if-te-1/0/2)# switchport
```

3. Enter the **switchport mode trunk** command to specify trunk mode.

```
device(conf-if-te-1/0/2)# switchport mode trunk
```

4. Enter the **switchport trunk allowed vlan add** command to add user-specified VLANs to the interface.

```
device(conf-if-te-1/0/2)# switchport trunk allowed vlan add 6000 ctag 10
```

5. Enter the **protected-port enable** command to enable the port as protected.

```
device(conf-if-te-1/0/2)# protected-port enable
```

6. Enter the **show protected-ports** command in the current configuration mode to confirm the configuration.

```
device(conf-if-te-1/0/1)# do show protected-ports
Vlan      Uplink Ports      Protected Ports
-----
10        TE1/0/1            TE1/0/2, TE1/0/3, TE1/0/4
6000     TE1/0/1, TE1/0/8  TE1/0/2, TE1/0/3, TE1/0/4, TE1/0/6
```

The **show vlan brief** command provides additional information.

VXLAN Overlay Gateways for NSX Controller Deployments

- Introduction to VXLAN overlay gateways with NSX Controller..... 67
- VXLAN NSX replicator load balancing..... 68
- Configuring a VXLAN overlay gateway for NSX Controller deployments..... 68
- Additional commands for VXLAN configuration..... 77

Introduction to VXLAN overlay gateways with NSX Controller

Virtual Extensible LAN (VXLAN) is an overlay network that extends Layer 2 domains over Layer 3 networks. The overlay network supports elastic compute architectures, enabling network engineers to scale a cloud computing environment while logically isolating cloud applications and tenants.

VXLAN extends the virtual LAN (VLAN) address space by adding a 24-bit segment ID and increasing the number of available VLAN IDs to 16 million, in a Virtual Fabrics context. The VXLAN segment ID in each frame differentiates individual logical networks, allowing millions of isolated Layer 2 VXLAN networks to coexist on a common Layer 3 infrastructure. As with VLANs, only virtual machines (VMs) within the same logical network can communicate with each other.

VXLAN creates large-scale, isolated virtual Layer 2 networks for virtualized and multi-tenant environments by encapsulating frames in VXLAN packets. Frame encapsulation is performed by means of a VXLAN Network Identifier (VNI) tunnel endpoint (VTEP), which originates or terminates VXLAN tunnels.

Because not all devices and servers are capable of sending or receiving VXLAN traffic, a VXLAN overlay gateway allows communication between the VXLAN-aware world and the non-VXLAN-aware world. In the non-VXLAN-aware world, a broadcast domain represented by a VLAN typically comprises the virtual cluster switch and other switches and devices behind the switch.

In the initial phase, the VXLAN-aware world consists of virtual networks that are managed by a third-party system known as the VMware NSX Controller. The NSX Controller is a highly available distributed system that manages, or orchestrates, all network components and connections in a virtual network. The VXLAN overlay gateway must communicate with the NSX Controller to create tunnels with VXLAN-aware end devices. The NSX Controller function can comprise a cluster of controllers. The orchestrator function resides most commonly at top of rack (ToR); however, it can also be deployed as an aggregator.

Beginning with Network OSv5.0.0, MAC, IPv4, and IPv6 ingress ACLs are supported, as well as sFlow configurations.

NOTE

Note the following conditions for this feature:

- VXLAN overlay gateways are supported only on the Extreme VDX 6740 family and VDX 6940 family.
- VXLAN gateways must be in logical chassis cluster mode. This allows the virtual cluster switch to present itself as a single device to the NSX Controller, in conjunction with a VMware Hypervisor.

Beginning with Network OSv7.0.0, support is provided for the following:

- Redistribution (load balancing) of broadcast, unknown unicast, and multicast (BUM) VLANs across VXLAN NSX Service Node (SN) tunnels in a VCS Fabric. (Service Nodes are now referred to as *replicators*.)
- VLAN classification profiles in hardware.

Beginning with Network OSv7.0.1, a default VLAN classification profile is provided to support NSX Controller deployments.

VXLAN NSX replicator load balancing

This feature enables the distribution of egress broadcast, unicast, and unknown multicast (BUM) traffic across tunnels to VMware NSX replicators, previously referred to as *Service Nodes (SNs)*.

When there are multiple replicator tunnels, the switch can distribute the BUM traffic across all such tunnels. This is done by assigning different set of VLANs to each of the tunnels. Egress BUM traffic on different VLANs are now forwarded through different tunnels, hence increasing the overall efficiency of the network. The switch can receive ingress BUM traffic over any VLAN on any replicator tunnel. When a tunnel is deleted or a Bidirectional Forwarding Detection (BFD) "down" state is detected, the system automatically reassigns its BUM VLANs to other available replicator tunnels.

Note the following considerations and limitations:

- This feature is supported on the Extreme VDX 6740 and VDX 6940 series. It is not supported on the Extreme VDX 8770 series.
- This feature is applicable only to NSX VXLAN tunnels, not to VXLAN overlay gateway L2-Extension VXLAN tunnels.
- BFD must be enabled on the NSX replicator.
- Momentary traffic disruptions can occur when new replicator tunnels come online. A system reboot is not required.
- Duplicate traffic can occur momentarily as replicator tunnels come online or go offline.
- VLANs can become "skewed" with respect to load balancing when the user removes specific ranges of VLANs from a given tunnel.
- You can manually trigger the redistribution of BUM VLANs on all NSX replicator tunnels by using the **tunnel replicator bum-vlans redistribute** command in privileged EXEC mode. Refer to the *Extreme Network OS Command Reference* for details on this command.
- This configuration cannot be changed if there are any replicator tunnels in the system. Because there is no option to remove only replicator tunnels, all tunnels must be removed, as follows: (1) remove all R Bridges attached to the overlay gateway, (2) delete the overlay gateway configuration, and (3) delete the NSX Controller configuration.
- Replicator load balancing is enabled following a firmware upgrade from Network OS 6.x or 7.0.0 to Network OS 7.0.1. A "disabled" load-balancing configuration is lost during a downgrade from Network OS 7.0.1, and the feature is lost during a downgrade to Network OS 6.x.

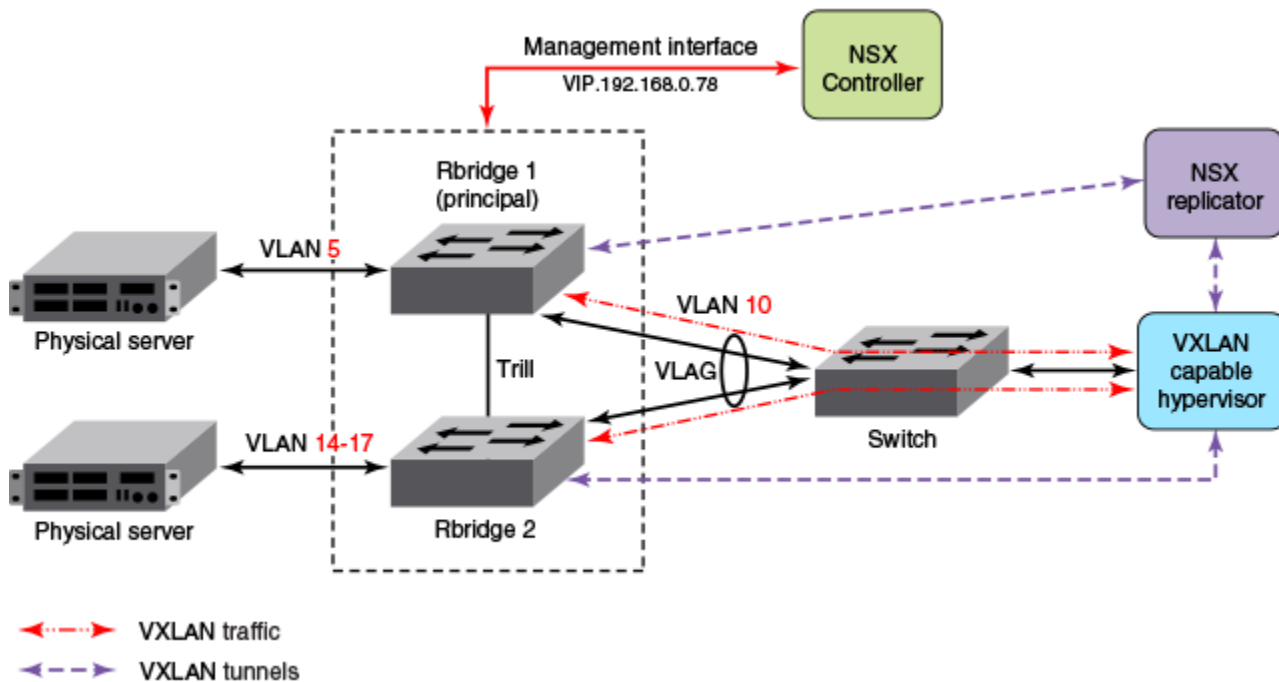
Configuring a VXLAN overlay gateway for NSX Controller deployments

Before you configure the VXLAN overlay gateway, complete the configuration steps in [Configuring VRRP-E for NSX Controller deployments](#) on page 70. The prerequisite steps demonstrate how to configure the R Bridges as part of a Virtual Router Redundancy Protocol - Extended (VRRP-E) group, which is a requirement for a VXLAN overlay gateway. Also, you must configure the identical virtual Ethernet (VE) and VRRP-E group on all the R Bridges for the VXLAN overlay gateway.

High-level communication in a VXLAN environment with an NSX Controller

The following illustration provides a basic view of the interaction of components in a VXLAN environment that uses an NSX Controller.

FIGURE 10 High-level communication for VXLAN overlay gateway with NSX Controller



A maximum of four Rbridges are supported in a VXLAN-enabled VCS Cluster. The VXLAN Gateway option should be enabled on all the Rbridges of the cluster.

In the example topology shown above, RBridge 1 and RBridge 2 make up a two-node cluster with the VXLAN gateway function enabled on both Rbridges. The Rbridges in the VCS cluster are connected to a VXLAN-capable hypervisor through a Layer 2 device. The VXLAN tunnel traffic is transported over VLAN 10 between the VCS and the VXLAN-capable hypervisor. An NSX replicator communicates with Rbridges through VXLAN tunnels. Additionally, there are two physical servers connected to the VCS cluster. The VXLAN gateway-enabled Rbridges transmit the VXLAN traffic from the hypervisor, as well as the VLAN-based traffic from the physical servers.

For the control communication, the principal switch of the VXLAN overlay gateway communicates with the NSX Controller. This communication occurs over the management interface (depicted by the red line in the illustration above). In this example, RBridge 1 is the principal RBridge for the VCS cluster.

Coordination of activities in NSX Controller deployments

Be sure to coordinate your activities with the administrators of the virtual network and NSX Controller to help ensure a successful setup. This includes providing the NSX administrator with an inventory of switches, ports, and VLANs in your cluster.

The NSX administrator creates the virtual network, assigns a VXLAN Network Identifier (VNI) to this network, and selects the ports that are to be attached to this virtual network. If ports on the virtual cluster switch are selected, the NSX Controller pushes network information to the switch. This information includes VNI, VLAN-to-VNI bindings for each port, and MAC-VNI-VTEP mappings for each of the MAC addresses in the virtual network. The NSX Controller is not aware of MAC addresses in the Layer 2 network behind the switch at this time.

Configuring VRRP-E for NSX Controller deployments

The steps that follow show an example VRRP-Extended (VRRP-E) group configuration for the R Bridges shown in [Figure 10](#) on page 69. VRRP-E is required to be configured on all the R Bridges of the VCS based VXLAN gateway for NSX.

NOTE

VRRP-E is necessary as the VXLAN tunnel termination and redundancy are related to the VRRP-E vMAC on the VDX 6740.

In the example shown in [Figure 10](#) on page 69, the VRRP-E functionality is configured on the VE interface for transport VLAN 10, which carries the VXLAN traffic between the VCS cluster and the VXLAN-capable hypervisor on each of the R Bridges in the VCS cluster.

NOTE

The existence of a VTEP for VXLAN bridging does not affect any configured routing and switching performed by the R Bridges in the VCS cluster for non-VXLAN traffic.

1. Enter global configuration mode:

```
switch# config
```

2. Enter RBridge ID configuration mode for the first RBridge in the logical chassis cluster (this example uses RBridge ID 1, as in the example topology).

```
switch(config)# rbridge-id 1
```

3. Enable VRRP-E for this RBridge.

```
switch(config-rbridge-id-1)# protocol vrrp-extended
```

4. Enter the **interface ve** command to configure a virtual Ethernet (VE) interface for RBridge 1 (for example, 10) that corresponds to an already created VLAN, and enter the IP address and mask for the interface (for example, 10.60.60.3/24).

```
switch(config-rbridge-id-2)# interface ve 10
```

5. Enter the IP address and mask for the interface (for example, 10.10.10.3/24).

```
switch(config-Ve-10)# ip address 10.10.10.3/24
```

6. Enter the **no shutdown** command to enable the interface.

```
switch(config-Ve-10)# no shutdown
```

7. Enter the **vrrp-extended-group** command for the group ID (100 in this example) of the VRRP-E group.

```
switch(config-Ve-10)# vrrp-extended-group 100
```

8. Assign a virtual MAC address by entering the **virtual-mac** command.

```
switch(config-vrrp-extended-group-100)# virtual-mac 02e0.5200.00xx
```

9. Enter a virtual IP address of the VRRP-E group, as in the following example.

```
switch(config-vrrp-extended-group-100)# virtual-ip 10.10.10.230
```

10. Enable short-path forwarding on the virtual router.

```
switch(config-vrrp-extended-group-100)# short-path-forwarding
```

- Exit this configuration mode and enter global configuration mode.

```
switch(config-vrrp-extended-group-100)# end
switch# configure
```

- Enter RBridge ID configuration mode for the other RBridge in your logical chassis cluster (RBridge 2 in the example topology).

```
switch(config)# rbridge-id 2
```

- Enable VRRP-E for this RBridge.

```
switch(config-rbridge-id-2)# protocol vrrp-extended
```

- Enter the **interface ve** command to configure the same VE interface as for RBridge 1.

```
switch(config-rbridge-id-2)# interface ve 10
```

- Enter the IP address and mask for RBridge 2 for this VE interface (for example, 10.60.60.4/24).

```
switch(config-Ve-10)# ip address 10.10.10.4/24
```

- Enter the **no shutdown** command to enable the interface.

```
switch(config-Ve-10)# no shutdown
```

- Enter the **vrrp-extended-group** command with the group ID of the VRRP-E group used on the other RBridge.

```
switch(config-Ve-10)# vrrp-extended-group 100
```

- Enter the **virtual-mac** command and assign the virtual MAC address used on RBridge 1.

```
switch(config-vrrp-extended-group-100)# virtual-mac 02e0.5200.00xx
```

- Enter a virtual IP address for the VRRP-E group, as in the following example.

```
switch(config-vrrp-extended-group-100)# virtual-ip 10.10.10.230
```

- Enable short-path forwarding on the virtual router.

```
switch(config-vrrp-extended-group-100)# short-path-forwarding
```

Configuring a loopback interface VTEP for NSX Controller deployments

As an alternative to the VRRP-E method, you can configure a loopback interface to serve as a VTEP.

Do the following to configure a loopback interface as a VTEP.

NOTE

You must manually configure distinct router IDs, by means of the **ip router-id** command, for use by routing protocols.

- Enter VXLAN overlay gateway configuration mode.

```
switch(config)# overlay-gateway gateway1
```

- Use the **ip interface** command to specify a loopback interface.

```
switch(config-overlay-gw-gateway1)# ip interface loopback 25
```

NOTE

When a VXLAN gateway is active (as configured by means of the **activate** command in VXLAN overlay gateway configuration mode), the loopback interface cannot be deleted. You must first use the **no activate** command.

VXLAN gateway and NSX Controller deployments

Consider the following guidelines before configuring the VXLAN gateway.

The following rules apply to VXLAN traffic packets transmitted by VXLAN-to-VLAN bridging:

- If the VXLAN packet entering a VDX VTEP-enabled device on the VLAN for which the VRRP-E session is addressed to the VRRP-E virtual MAC address, and the final destination is an NSX-configured VXLAN tunnel (as identified by the tunnel parameters of source IP and destination IP addresses), then the VXLAN traffic is a candidate for VXLAN-to-VLAN bridging.
- If the VXLAN packet entering a VDX VTEP-enabled device on a Layer 3 interface (such as a routing next hop) that is different from the VRRP-E-based virtual Ethernet (VE) interface configured for the VTEP, and its final destination is an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of source IP and destination IP addresses), then the VXLAN traffic is routed to the VTEP interface in the Extreme VDX device and is a candidate for VXLAN-to-VLAN bridging.
- If the VXLAN packet entering a VDX VTEP-enabled device on a Layer 3 interface (such as a routing next hop) is different from the VRRP-E-based VE interface configured for the VTEP, but is at an ingress interface where the VTEP VRRP-E VLAN is also configured, and the final destination is an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of source IP and destination IP addresses), then the VXLAN traffic is routed to the VTEP interface in the Extreme VDX device and is a candidate for VXLAN-to-VLAN bridging, but only if the destination MAC address of the ingressing VXLAN traffic is the same as that of the virtual MAC address of the VTEP VRRP-E session. This occurs if the user creates VE interfaces for each of the ingressing transport VLANs on all the RBridges in the VCS Fabric, and then configures them with the same VRRP-E VRID and virtual MAC address as the VRRP-E VRID and the virtual MAC address that was configured for the VTEP.

Configuring the VXLAN gateway for NSX Controller deployments

The following steps detail how to configure a VXLAN overlay gateway and point it to the NSX Controller. This procedure references [Figure 10](#) on page 69. Both the NSX for Multi-Hypervisor (NSX-MH) and the NSX for vSphere (NSX-V) are supported. The configuration for an NS-MH requires TCP port 6632 to support the Open vSwitch Database Management Protocol (OVSDB) management channel. The configuration for an NSX-V requires TCP port 6640 to support the OVSDB management channel.

Perform all the following steps on the principal switch (RBridge 1 in the example topology).

NOTE

A tunnel will not be created if there is not an active VM on the Hypervisor. If the tunnel is not created, check the VM connectivity on the Hypervisor.

1. The following substeps create a VXLAN Network Identifier (VNI) tunnel endpoint (VTEP).

- a) Enter global configuration mode, then enter the **overlay-gateway name type hardware-vtep** command.

NOTE

The **type hardware-vtep** keywords are required to specify that this deployment uses an NSX Controller (this keyword also supports OpenStack deployments. The name "gateway1" is only an example; this can be a name of your choice.

```
device# config
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)# type hardware-vtep
```

- b) Enter the **attach rbridge-id** command to attach existing RBridge IDs to this VXLAN gateway instance. You can specify a range of RBridge IDs up to a maximum of four.

```
device(config-overlay-gw-gateway1)# attach rbridge-id add 1-2
```

- c) Enter the **ip interface ve veid vrrp-extended-group group-ID** command to set the IP address of the VXLAN overlay gateway, as shown in the following example.

```
device(config-overlay-gw-gateway1)# ip interface ve 10 vrrp-extended-group
100
```

The command accepts the VE interface ID and VRRP-E group ID, then sets the VXLAN overlay gateway's IP address as identical to the already configured VRRP-E virtual IP address. Tunnels that form use this IP address as the source IP address for outgoing packets.

- d) Enter the **attach vlan vlan_id** command to export the desired VLANs (these are VLANs that can be mapped to VXLAN domains), as shown in the following example.

```
device(config-overlay-gw-gateway1)# attach vlan 5,14-17
```

NOTE

Virtual Fabrics cannot be attached when the overlay gateway type is **hardware-vtep** and Virtual Fabrics cannot be extended.

All the MAC addresses that the VXLAN overlay gateway learns on these VLANs are shared with the NSX Controller. When a MAC address ages out in VCS, the MAC address is removed from the NSX Controller.

NOTE

There is also an option to list specific MAC addresses. In this case, other MAC addresses that are learned for the VLAN are not shared with the NSX Controller. For more information, refer to the **attach vlan** command in the *Network OS Command Reference*.

- e) Optional: (Optional) You can enter the **enable statistics direction** command to enable statistics collection for tunnels you specify, as shown in the following example.

```
device(config-overlay-gw-gateway1)# enable statistics direction both vlan
add 14-17
```

This example command enables statistics collection for tunnels in both directions (transmitting and receiving) for the specified VLANs.

- f) Optional: If you have created a SPAN destination outside of the VXLAN overlay gateway (as a monitor session), you can enter the **monitor session** command to monitor session traffic, as shown in the following examples.

```
device(config-overlay-gw-gateway1)# monitor session 1 direction both
remote-endpoint 1.2.3.4 vlan add 41-43
```

```
device(config-overlay-gw-gateway1)# monitor session 1 direction both
remote-endpoint any vlan add 41-43
```

- g) Enter the **activate** command to activate this gateway instance.

```
device(config-overlay-gw-gateway1)# activate
```

This enables all tunnels associated with this gateway. VXLAN tunnels are not user configurable.

- h) Return to privileged EXEC mode.

```
device(config-overlay-gw-gateway1)# end
device(config)# end
device#
```

2. Generate the security certificate for the VXLAN overlay gateway by entering the **nsx-controller client-cert** command with the **generate digest** keywords.

NOTE

Beginning with Network OS 7.1.0, support is provided for SHA-2 family cryptographic hash functions.

ATTENTION

The SHA-2 family is supported only for Open vSwitch Management Database (OVSDB) connections.

```
device# nsx-controller client-cert generate digest sha256
```

Certificate generation is a one-time-only action. Use the **nsx-controller client-cert delete** command to delete a previously generated certificate.

3. In privileged EXEC mode, display the certificate by entering the **show nsx-controller client-cert** command, then provide the certificate to the NSX administrator.

4. The following substeps configure the management interface (depicted by the red line in [Figure 10](#) on page 69), which allows communication between the VXLAN gateway and the NSX Controller:

- a) Enter global configuration mode.

```
device# config
```

- b) Enter the **vcs virtual ip address** command and assign an IP address and mask.

```
device(config)# vcs virtual ip address
192.168.0.78/24
```

- c) Enter the **nsx-controller name** command to specify a name for a new NSX Controller connection profile:

```
device(config)# nsx-controller profile1
```

- d) Enter the **ip address** command to set the IP address of the controller, the port, and connection-method settings for an NSX Controller connection profile as shown in this example.

NOTE

This example illustrates a configuration for the NSX-MH, which requires port 6632 to support the OVSDB management channel. Use port 6640 when configuring support for an NSX-V.

```
device(config-nsx-controller-profile1)# ip address 10.21.83.188 port 6632
```

- e) Optional: You can change the reconnect interval between the NSX Controller and the VCS Fabric in case the connection is lost. The default is 10 seconds, meaning that a reconnection is attempted every 10 seconds. To change this interval to 40 seconds, for example, use the **reconnect-interval** command:

```
device(config-nsx-controller-profile1)# reconnect-interval 40
```

- f) Finally, enter the **activate** command to activate the NSX Controller profile.

```
device(config-nsx-controller-profile1)# activate
```

This command initiates the connection between the NSX Controller and the VCS Fabric.

NOTE

The following rules apply to a VXLAN packet entering an interface on a VTEP-enabled Extreme VDX switch for that packet to be a candidate for VXLAN-to-VLAN bridging:

- If the VXLAN packet is entering a VTEP-enabled switch on the VLAN for which the VRRP-E session is configured, and the packet is destined to the VRRP-E virtual MAC address and belongs to an NSX-configured VXLAN tunnel (as identified by the source and destination IP address in the VXLAN packet), then the VXLAN packet is a candidate for VXLAN-to-VLAN bridging.
- If the VXLAN packet is entering a VTEP-enabled switch at a Layer 3 interface (as a routing next hop) that is different from the VRRP-E based VE interface configured for the VTEP, and the packet belongs to an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of the source and destination IP addresses), then the VXLAN traffic is routed to the VTEP interface in the Extreme VDX and is a candidate for VXLAN-to-VLAN bridging, with the exception noted below.
- If the VXLAN packet entering a VTEP-enabled switch at a Layer 3 interface (as a routing next hop) that is different from the VRRP-E-based VE interface configured for the VTEP but is an ingress interface where the VTEP VRRP-E VLAN is also configured, and the packet belongs to an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of the source and destination IP addresses), then the VXLAN traffic is routed to the VTEP interface in the Extreme VDX and is a candidate for VXLAN-to-VLAN bridging, only if the destination MAC address of the ingressing VXLAN packet is the same as that of the virtual MAC

address of the VTEP VRRP-E session. This use case is addressed by creating VE interfaces for each of the ingressing transport VLANs on all the Rbridges in the VCS Fabric and configuring them with the same VRRP-E VRID and virtual MAC address as the VRRP-E VRID and virtual MAC address configured for the VTEP.

Configuring VXLAN NSX replicator load balancing

You can manually trigger the redistribution of broadcast/unicast/multicast (BUM) VLANs on all NSX replicator tunnels. This feature is supported on the Extreme VDX 6740 and VDX 6940 series. It is not supported on the Extreme VDX 8770 series.

For details, see the section "VXLAN NSX replicator load balancing" in this chapter.

1. In privileged EXEC mode, enter the **tunnel replicator bum-vlans redistribute** command.

```
device# tunnel replicator bum-vlans redistribute
```

2. To view details of BUM-enabled replicator tunnels, as well as the BUM VLANs (in range format) for each tunnel, use the **show tunnel replicator** command as in the following example:

```
device# show tunnel replicator
Tunnel 61442, mode VXLAN, rbridge-ids 1
Ifindex 2080436226, Admin state up, Oper state up, BFD up
Overlay gateway "GW1", ID 1
Source IP 20.20.1.1 ( Loopback 11), Vrf default-vrf
Destination IP 20.20.0.197
Configuration source VTEP Controller
MAC learning disabled
BUM vlans 41,1414-1813 (401 vlans)
Active next hops on rbridge 1:
  IP: 19.1.0.2, Vrf: default-vrf
  Egress L3 port: Te 1/0/15, Outer SMAC: 0027.f8db.e068
  Outer DMAC: 0027.f83a.349d
  Egress L2 Port: Te 1/0/15, Outer ctag: 0, stag:0, Egress mode: Local
  BUM forwarder: no

Packet count: RX 11441          TX 0
Byte count   : RX (NA)         TX 0

Tunnel 61443, mode VXLAN, rbridge-ids 1
Ifindex 2080436227, Admin state up, Oper state up, BFD up
Overlay gateway "GW1", ID 1
Source IP 20.20.1.1 ( Loopback 11), Vrf default-vrf
Destination IP 20.20.0.181
Configuration source VTEP Controller
MAC learning disabled
BUM vlans 42,1814-2213 (401 vlans)
Active next hops on rbridge 1:
  IP: 19.1.0.2, Vrf: default-vrf
  Egress L3 port: Te 1/0/15, Outer SMAC: 0027.f8db.e068
  Outer DMAC: 0027.f83a.349d
  Egress L2 Port: Te 1/0/15, Outer ctag: 0, stag:0, Egress mode: Local
  BUM forwarder: no

Packet count: RX 11436          TX 0
Byte count   : RX (NA)         TX 0
```

Additional commands for VXLAN configuration

Additional commands that support VXLAN configuration are listed in the following table.

NOTE

For complete information on the those commands as well as other VXLAN overlay-gateway commands and commands related to the NSX Controller, refer to the *Network OS Command Reference*.

TABLE 6 Additional commands for VXLAN configuration

Command	Description
clear overlay-gateway	Clears counters for the specified gateway.
enable statistics	Enables statistics for tunnels.
sflow remote-endpoint	Applies an sFlow profile for a VXLAN overlay gateway and sets the remote endpoints for tunnel interfaces.
show nsx controller	Displays connection status of the NSX Controller. Includes an option to display the gateway certificate that is needed for NSX "transport node" configuration.
show overlay-gateway	Displays status and statistics for the VXLAN overlay-gateway instance.
show running-config overlay-gateway	Displays the running configuration of the overlay gateway configuration, including the connection type.
show tunnel	Displays tunnel statistics, including those for the NSX Service Node.

Distributed VXLAN Gateways

- [Distributed VXLAN gateways overview.....](#)79
- [Configuring a distributed VXLAN gateway.....](#)84
- [Troubleshooting and managing distributed VXLAN gateways.....](#)85

Distributed VXLAN gateways overview

The distributed VXLAN gateways feature eliminates the need for an external gateway device.

Prior to Network OS 6.0.1, VXLAN gateways had to be connected to the VCS Fabric (for example, a four-node gateway fabric) as an external device in order to bridge the overlay network and the physical networks that are interconnected by the VCS Fabric. Beginning with the current release, the gateway function can be hosted by the VCS R Bridges. This eliminates the need for an external gateway device and optimizes network resources, improving network performance in the data center.

NOTE

Existing VRRP-E implementations cannot support more than four R Bridges in a session. As a result, VRRP-E-based VXLAN gateways are also limited to a maximum of four R Bridges.

The following sections present various use cases, both supported and unsupported, and their corresponding topologies. Refer to the following legend for those topologies.

FIGURE 11 Distributed VXLAN gateways legend



NOTE

Distributed VXLAN gateways support both Virtual Fabrics Extension deployments and NSX Controller deployments.

Distributed VXLAN gateways supported topologies

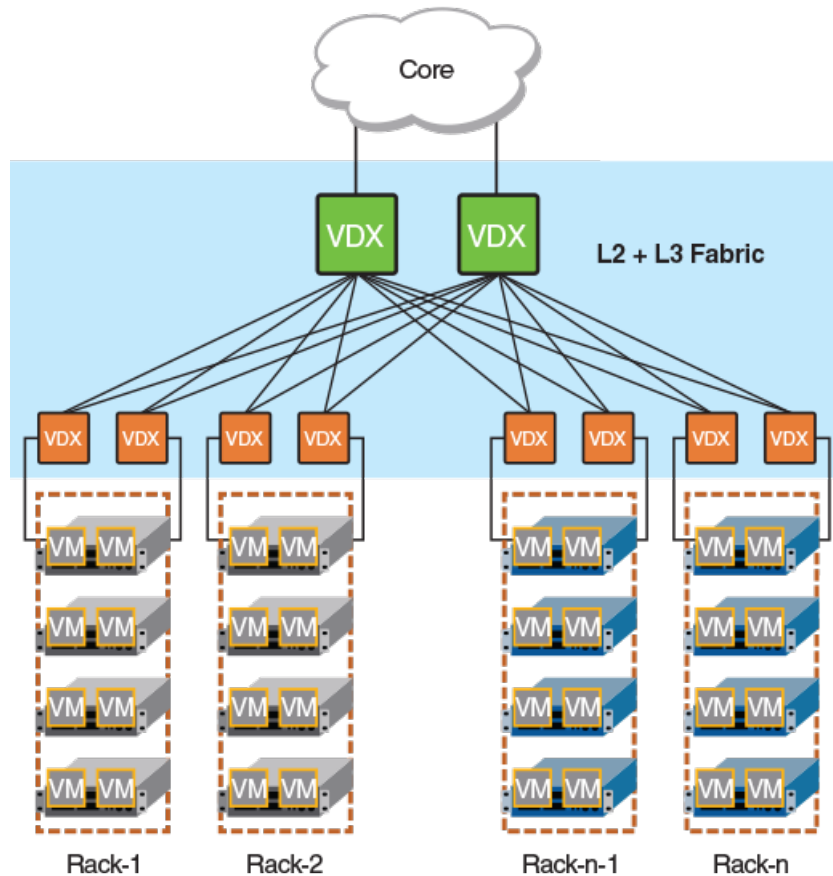
The following sections present the topologies that are supported in this release.

VDX 6940 at the aggregation layer

An Extreme VDX 6940 at the aggregation layer provides gateway functions for a VXLAN hypervisor at top of rack (ToR) or in the core.

A distributed VXLAN gateway makes it possible to connect to a hypervisor through a TRILL fabric, as the Extreme VDX 6940 supports TRILL-plus-VXLAN encapsulation. Refer to the following figure.

FIGURE 12 VDX 6940 in a Layer 2/Layer 3 fabric



In this topology, bridging a physical server and VXLAN servers that are located in the same rack requires interaction with an aggregation gateway, introducing an extra hop. The gateway supports VXLAN Network Identifier (VNI) classification for both east-west and north-south traffic. Note the following considerations:

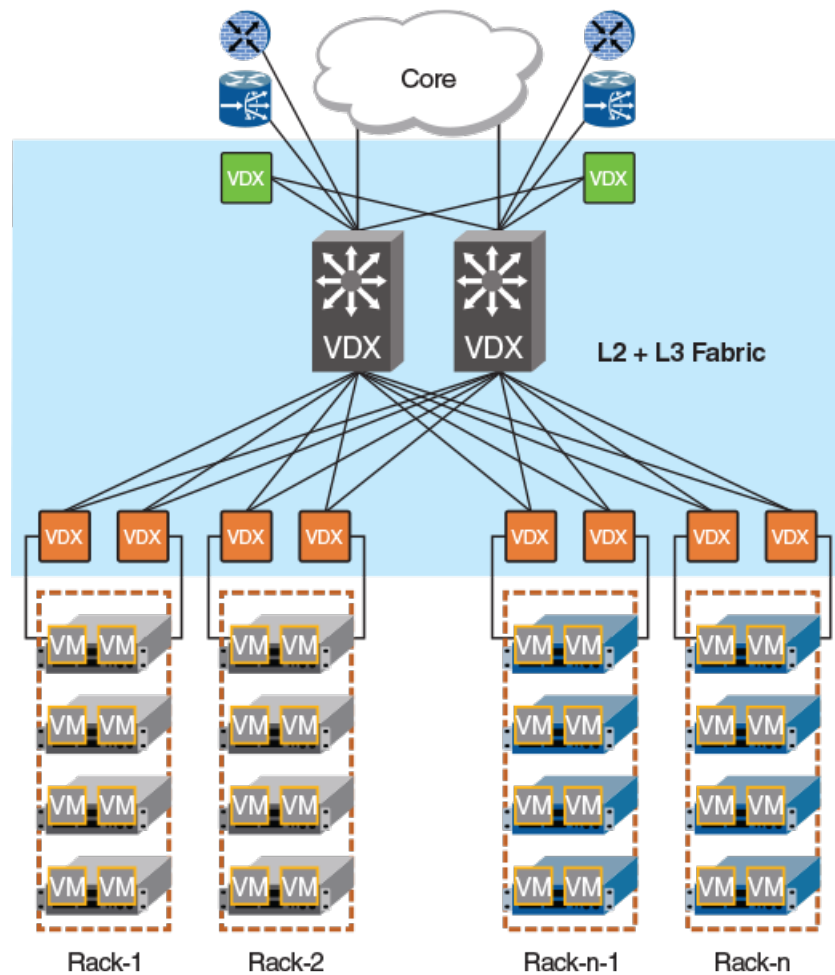
- The number of overlay networks that are supported in the fabric is limited by ASIC resources for the VNI classifications.
- The Extreme VDX 6940 is not supported at top of rack. This prevents VLAN-to-VXLAN traffic from "tromboning" in case the ToR gateway that does the VXLAN encapsulation is not in the same rack that holds the VXLAN server.

VDX 6940 as an appliance

An Extreme 6940 gateway can be inserted into the fabric as an appliance.

The following figure illustrates how a data center can use a VXLAN-incapable aggregation switch (such as an Extreme VDX 8770) to provide connectivity to the core while the switch is attached to a VXLAN gateway functioning as an appliance. Traffic between the physical server and a VXLAN-enabled server must always take an extra hop through the VDX to reach the appliance gateway.

FIGURE 13 VDX 6940 as a gateway appliance



Distributed VXLAN gateways unsupported topologies

The following topologies, although they can provide a certain degree of functionality, are not supported by Extreme for this feature.

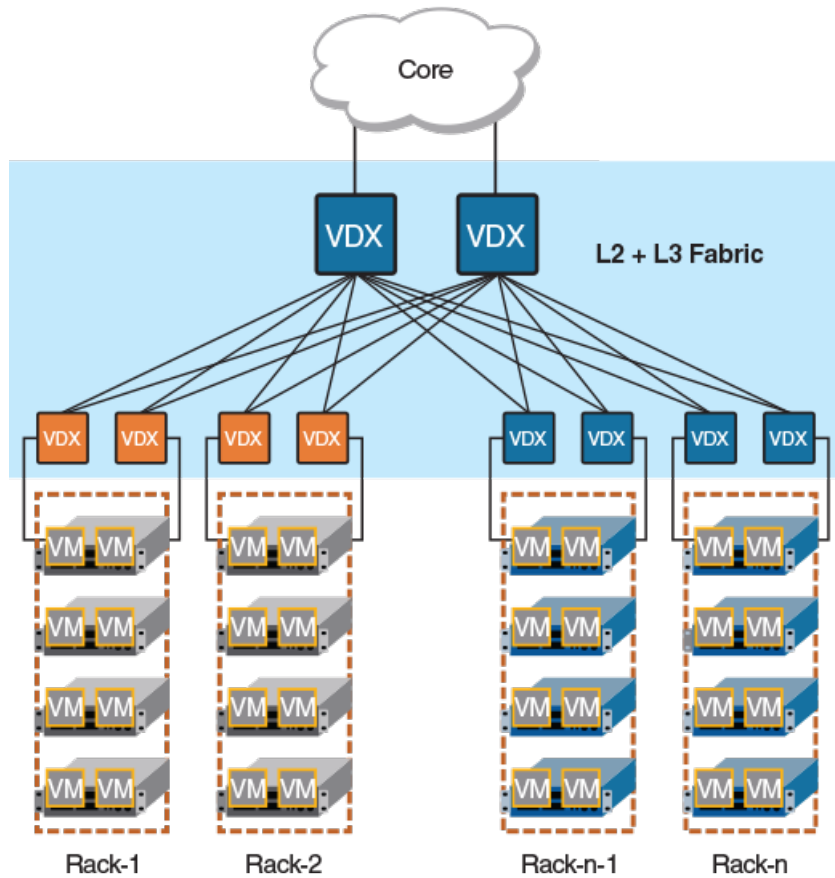
VDX 6740-based fabric

In an Extreme VDX 6740-based fabric, the gateway functionality is distributed across every VDX 6740 RBridge.

This topology is not recommended because of the following limitations:

- It requires a direct attachment between the gateway and a VXLAN-enabled rack, because the VDX-6740 cannot support TRILL-plus-VXLAN encapsulation.
- The number of server racks is limited to eight. This is constrained by the maximum number of R Bridges that are allowed in a gateway.
- The VDX 6740 supports a maximum of 2000 VLANs.

FIGURE 14 VDX 6740-based fabric



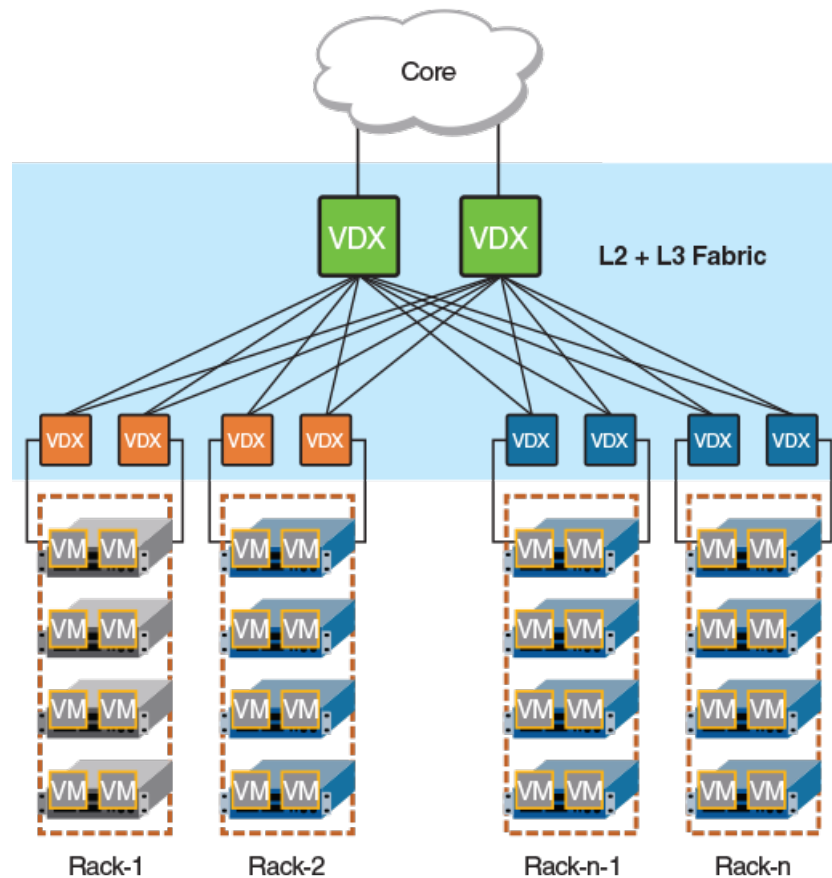
VDX 6740 and VDX 6940-based fabric

In this topology, a gateway comprises a mix of Extreme VDX 6740 and VDX 6940 R Bridges.

The VDX 6740 is placed at top of rack to serve directly connected VXLAN servers. The VDX 6740 gateway handles VNI classifications for east-west traffic, thereby offloading traffic to the VDX 6940 gateway at the aggregation layer to serve other north-south traffic classifications.

Although this topology helps scale out the number of overlay networks supported in the fabric, it has the same limitations as in the previous topology, where the VDX 6740 is placed at top of rack.

FIGURE 15 VDX 6740 and VDX 6940-based fabric



Distributed VXLAN gateways RBridge scalability

Every RBridge in the fabric can act as a gateway. However, because dynamic virtual RBridge IDs (VRBs) support a maximum of eight RBridges, a tunnel VRB-ID can represent a maximum of eight RBridges out of all the gateway RBridges that are specified in the overlay-gateway configuration.

NOTE

The maximum number of gateway RBridges deployed at the aggregation layer is four. If this number is exceeded, the RBridges that are reachable by means of VRBs is nondeterministic.

Distributed VXLAN gateways upgrade and downgrade considerations

For the Extreme VDX 6740, after an upgrade or downgrade, the existing four-node VDX 6740 fabric can continue to serve as a four-node gateway fabric, but it should not join another fabric. In addition, the Virtual Fabrics extension gateway at the aggregation layer can continue to act as an extension gateway, but it cannot be configured as a VXLAN gateway.

ATTENTION

In a downgrade from the current release to a release earlier than Network OS 6.0.1, any new commands introduced in this release must be removed from devices before the downgrade is honored. Also, TRILL+VXLAN functionality is lost during a downgrade, and there is no warning to the user.

Distributed VXLAN gateways limitations

The following functions are not supported for distributed VXLAN gateways:

- BUM optimization
- Loop detection that involves both a tunnel and a nontunnel path
- Flow-based load balancing for tunnels over router ports
- Routing protocols over tunnels
- More than one VTEP per fabric
- QoS that is limited to DiffServ tunneling pipe mode
- A SPAN destination that is not a tunnel

Configuring a distributed VXLAN gateway

This task uses the **overlay-gateway** command and related commands to configure a distributed VXLAN gateway.

1. Enter global configuration mode.

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

2. Enter the **overlay-gateway** command and specify a gateway, entering VXLAN overlay gateway configuration mode.

```
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)#
```

3. Enter the **type** command to specify the gateway type as Layer 2 extension.

```
device(config-overlay-gw-gateway1)# type layer2-extension
```

4. Enter the **ip interface** command to specify Loopback 1 as the IPv4 interface.

```
device(config-overlay-gw-gateway1)# ip interface loopback 1
```

5. Enter the **attach rbridge-id** command to attach R Bridges as appropriate for your network, as in the following example.

```
device(config-overlay-gw-gateway1)# attach rbridge-id add 3,4
```

6. Enter the **map vlan vni** command with the **auto** keyword to enable automatic VLAN-to-VNI mapping for every VLAN associated with the tunnel.

```
device(config-overlay-gw-gateway1)# map vlan vni auto
```

7. Enter the **site** command to create a remote Layer 2 extension site in a VXLAN overlay gateway context and enable VXLAN overlay gateway site configuration mode.

```
device(config-overlay-gw-gateway1)# site site1
device(config-site-site1)#
```

8. Enter the **ip address (VXLAN)** command to specify the destination IPv4 address of a tunnel.

```
device(config-site-site1)# ip address 10.1.1.1
```

- Enter the **extend vlan** command to configure a switchport VLAN (or VLANs, as appropriate) for the tunnel to the containing site, as in the following example.

```
device(config-site-site1)# extend vlan add 1005
```

- Enter the **activate (VXLAN gateway)** command to activate the gateway, and return to VXLAN overlay gateway configuration mode.

```
device(config-site-site1)# activate
device(config-overlay-gw-gateway1)#
```

Troubleshooting and managing distributed VXLAN gateways

A variety of options are available for managing and troubleshooting distributed VXLAN gateways.

Troubleshooting

The following examples illustrate commands used to confirm VTEP and tunnel configurations, VLAN activation, and MAC addresses, as well as to view gateway statistics.

To confirm VTEP and tunnel configurations:

```
device# show running-config overlay-gateway
overlay-gateway gateway2
  type layer2-extension
  ip interface Ve 29 vrrp-extended-group 255
  attach rbridge-id add 1-2
  attach vlan 2
  activate

device# show tunnel brief rbridge-id 1
Tunnel 61441, mode VXLAN, rbridge-ids 10
Admin state UP, Oper state UP
Source IP 60.60.60.29, Vrf default-vrf
Destination IP 20.1.1.1

device# show tunnel 61441
Tunnel 61441, type nsx, rbridge-ids 1,2
Admin state UP, Oper state UP
Source IP 60.60.60.29, Vrf default-vrf
Destination IP 60.60.60.184

device# show tunnel statistics
Tnl ID   RX packets   TX packets   RX bytes   TX bytes
=====
61441   123          456          (NA)       4560
61442   567          890          (NA)       8900

device# show system internal tnlmgr tunnel all
```

To confirm VLAN activation:

```
device# show vlan brief
Total Number of VLANs configured      : 14
Total Number of VLANs provisioned     : 14
Total Number of VLANs unprovisioned   : 0
VLAN      Name                State                Ports                Classification
(F)-FCoE
(R)-RSPAN
(T)-TRANSPARENT
=====
1          default             ACTIVE              Po 333(t)            (u)-Untagged
Po 444(t)            (c)-Converged
Tu 61441(t)         (t)-Tagged
2          VLAN2                 ACTIVE              Tu 61441(t)         vni 3029)
1002(F)    VLAN1002                INACTIVE(no member port)
```

To confirm MAC addresses:

```
device# show mac-address-table
VlanId  Mac-address      Type      State      Ports
2       0000.0000.0001   Dynamic   Active     Tu 61441
2       0000.0000.0002   Dynamic   Active     Tu 61441
2       0000.0000.0003   Dynamic   Active     Tu 61441
2       0000.0000.0004   Dynamic   Active     Tu 61441
2       0000.0000.0005   Dynamic   Active     Tu 61441
2       0000.0000.0006   Dynamic   Active     Tu 61441
20      0027.f880.1890   System    Remote     XX 40/X/X
20      0027.f880.328f   Dynamic   Active     Po 1
```

To view gateway statistics:

```
device# show overlay-gateway name test vlan statistics
VLAN ID  RX packets      TX packets
=====
30       1234            5678
```

Enabling SPAN

You can use the Switched Port Analyzer (SPAN) protocol to monitor traffic by means of the **monitor session** command, as in the following examples:

```
device(config-overlay-gw-gateway3)# monitor session 1 direction both remote-endpoint 1.2.3.4 vlan add 2
device(config-overlay-gw-gateway3)# monitor session 1 direction both remote-endpoint any vlan add 2
```

NOTE

A SPAN destination must first be created outside the overlay gateway (as a monitor session). All regular SPAN sessions are supported.

STP-Type Protocols

- STP overview.....87
- Configuring and managing STP and STP variants..... 91
- Cisco Peer-Switch support..... 110

STP overview

A network topology of bridges typically contains redundant connections to provide alternate paths in case of link failures. However, because there is no concept of TTL in Ethernet frames, this could result in the permanent circulation of frames if there are loops in the network. To prevent loops, a spanning tree connecting all the bridges is formed in real time. The redundant ports are put in a blocking (nonforwarding) state. They are enabled when required. In order to build a spanning tree for the bridge topology, the bridges must exchange control frames (BPDUs - Bridge Protocol Data Units). The protocols define the semantics of the BPDUs and the required state machine. The first Spanning Tree Protocol (STP) became part of the IEEE 802.1d standard.

The STP interface states for every Layer 2 interface running STP are as follows:

- *Blocking* – The interface does not forward frames.
- *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.
- *Learning* – The interface prepares to participate in frame forwarding.
- *Forwarding* – The interface forwards frames.
- *Disabled* – The interface is not participating in spanning tree because of a shutdown port, no link on the port, or no spanning tree instance running on the port.

A port participating in spanning tree moves through these states:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding, blocking, or disabled
- From forwarding to disabled

The following STP features are considered optional features, although you might use them in your STP configuration:

- *Root guard*
- *Port fast BPDU guard* and *BPDU filter*

STP configuration guidelines and restrictions

Follow these configuration guidelines and restrictions when configuring STP:

- You have to disable one form of xSTP before enabling another.
- Packet drops or packet flooding may occur if you do not enable xSTP on all devices connected on both sides of parallel links.
- Network OS switches in logical chassis cluster mode drop all tagged xSTP frames that originate from an Extreme VDX.
- In the case of STP over VCS (STP over VCS), STP is disabled by default on all ports.

- When a misconfigured local area network running spanning tree has one or more loops, a traffic storm of spanning tree BPDUs can occur. In certain circumstances, VDX switches can reboot when subjected to an extended period of traffic storm involving spanning tree BPDUs.
- Additionally, when a misconfigured local area network running spanning tree has one or more loops, a traffic storm of spanning tree BPDUs can occur. Edge Loop Detection (ELD) protocol cannot eliminate loops during a traffic storm involving control packets, such as spanning tree BPDUs.
- For load balancing across redundant paths in the network to work, all VLAN-to-instance mapping assignments must match; otherwise, all traffic flows on a single link.
- Spanning tree topologies must not be enabled on any direct server connections to the front-end 10-gigabit Ethernet ports that may run FCoE traffic. This can result in lost or dropped FCoE logins.

RSTP

NOTE

Rapid Spanning Tree Protocol is designed to be compatible and interoperate with STP. However, the advantages of the RSTP fast reconvergence are lost when it interoperates with switches running STP.

The IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard is an evolution of the 802.1D STP standard. It provides rapid reconvergence following the failure of a switch, a switch port, or a LAN. It provides rapid reconvergence of edge ports, new root ports, and ports connected through point-to-point links.

The RSTP interface states for every Layer 2 interface running RSTP are as follows:

- *Learning* — The interface prepares to participate in frame forwarding.
- *Forwarding* — The interface forwards frames.
- *Discarding* — The interface discards frames. Note that the 802.1D disabled, blocking, and listening states are merged into the RSTP discarding state. Ports in the discarding state do not take part in the active topology and do not learn MAC addresses.

The following table lists the interface state changes between STP and RSTP.

TABLE 7 STP versus RSTP state comparison

STP interface state	RSTP interface state	Is the interface included in the active topology?	Is the interface learning MAC addresses?
Disabled	Discarding	No	No
Blocking	Discarding	No	No
Listening	Discarding	Yes	No
Learning	Learning	Yes	Yes
Forwarding	Forwarding	Yes	Yes

With RSTP, the port roles for the interface are also different. RSTP differentiates explicitly between the state of the port and the role it plays in the topology. RSTP uses the root port and designated port roles defined by 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.

MSTP

IEEE 802.1s Multiple STP (MSTP) helps create multiple loop-free active topologies on a single physical topology. MSTP enables multiple VLANs to be mapped to the same spanning tree instance (forwarding path), which reduces the number of spanning tree instances needed to support a large number of VLANs. Each MSTP instance has a spanning tree topology independent of other spanning tree instances. With MSTP you can have multiple forwarding paths for data traffic. A failure in one instance does not affect other instances. With MSTP, you are able to more effectively utilize the physical resources present in the network and achieve better load balancing of VLAN traffic.

NOTE

In MSTP mode, RSTP is automatically enabled to provide rapid convergence.

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.

NOTE

Network OS supports 32 MSTP instances and one MSTP region.

MSTP introduces a hierarchical way of managing switch domains using regions. Switches that share common MSTP configuration attributes belong to a region. The MSTP configuration determines the MSTP region where each switch 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 configuration. A multiple spanning tree instance 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 common internal spanning tree (CIST) that forms a single spanning tree instance that includes all the switches 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 RSTP if all the switches across the regions support RSTP. However, if any of the switches operate using 802.1D STP, the CIST instance reverts to 802.1D. Each region is viewed logically as a single STP/RSTP bridge to other regions.

MSTP guidelines and restrictions

Follow these restrictions and guidelines when configuring MSTP:

- You can have 32 MSTP instances and one MSTP region.
- Create VLANs before mapping them to MSTP instances.
- The MSTP **force-version** option is not supported.
- When you enable MSTP by using the **global protocol spanning-tree mstp** command, RSTP is enabled automatically.
- For two or more switches to be in the same MSTP region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same name.

PVST+ and Rapid PVST+

Both STP and 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. If a port is set to "blocked/discarding" for one VLAN (under STP/RSTP), it is the same for all other VLANs too.

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 switch.

PVST+ is not a scalable model when there are many VLANs in the network, as it consumes a lot of CPU power. A reasonable compromise between the two extremes of RSTP and R-PVST+ is the Multiple Spanning Tree protocol (MSTP), which was standardized as IEEE 802.1s and later incorporated into the IEEE 802.1Q-2003 standard. MSTP runs multiple instances of spanning tree that are independent of VLANs. It then maps a set of VLANs to each instance.

NOTE

Network OS 4.0 and later supports PVST+ and R-PVST+ only. The PVST and R-PVST protocols are proprietary to Cisco and are not supported.

PVST+ and R-PVST+ guidelines and restrictions

Consider the following when configuring PVST+ and R-PVST+:

- Disabling the tagging of native VLANs is required on STP/RSTP/MSTP VDX devices in standalone mode, otherwise PVST/ RPVST does not converge and forms a loop on the native VLAN. The tagged native VLAN data traffic is ignored. The native VLAN untagged data is forwarded.
- If a VLAN is configured with tagged ports that have RSTP mode enabled but not PVST+ mode enabled, and are connected to VDX devices, then BPDUs from the tagged ports that are received by the VDX device are dropped.

Spanning Tree Protocol and VCS mode

Network OS 4.0 and later supports any version of STP to run in VCS mode and function correctly between interconnecting VCS nodes, or between VCS and other vendor's switches. This feature is called Distributed Spanning Tree Protocol (DiST).

The purpose of DiST is as follows:

- To support VCS to VCS connectivity and automatic loop detection and prevention.
- To assist deployment plans for replacing the legacy xSTP enabled switches in your network.

DiST supports any of the following flavors of xSTP:

- IEEE STP
- IEEE RSTP
- IEEE MSTP
- PVST
- PVRST

DiST treats one VCS as one virtual xSTP bridge from an external view. Each VCS has a unique RBridge ID and Priority. DiST can be enabled on VCS edge ports connecting to other VCS nodes. The Port Ids used for xSTP are dynamically assigned and unique within the VCS.

Each RBridge runs the spanning tree instance in a distributed manner. Each spanning tree instance considers all the edge ports and the best information from the remote RBridges to arrive at the spanning tree topology. Each RBridge updates all the other members about its best information for a given spanning tree instance.

Each RBridge maintains a table of best information from the other RBridges in the cluster. This table is identical across all the RBridges in the cluster. This information is used to derive the port roles for the local edge ports. The shared information of the whole cluster is considered in the spanning tree calculations for port roles and states of local edge ports. Thus, all the remote RBridges' edge port information could affect the port role selection and port state transitions for the local edge ports. This ensures that each RBridge considers the port roles and states of all the other RBridges to arrive at a final spanning tree topology.

In the event of a change of the "best" information on any member RBridge, that RBridge would update its own next best information to the other RBridges. Some of the scenarios in which this could happen are the following:

- Operational status change of port associated with the "best" information
- Reception of superior information by another edge port on the RBridge
- Reception of superior or inferior information by the "best" port on the RBridge
- Nonreception of BPDUs on the best port for a given period of time

The xSTP update information is received by all member nodes of the cluster. Each node updates its internal database with the received information. If this results in a best-information change, the update is applied on to the logical port for the node. This triggers the xSTP state machine for all local ports.

Configuring and managing STP and STP variants

Before proceeding, refer to [STP configuration guidelines and restrictions](#) on page 87.

Understanding the default STP configuration

It is helpful to understand the STP defaults before you make configuration changes. The following table lists the default STP configuration.

TABLE 8 Default STP configuration

Parameter	Default setting
Spanning-tree mode	By default, STP, RSTP, and MSTP are disabled
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 those switch defaults which apply only to MSTP configurations.

TABLE 9 Default MSTP configuration

Parameter	Default setting
Cisco interoperability	Disabled

TABLE 9 Default MSTP configuration (continued)

Parameter	Default setting
Switch priority (when mapping a VLAN to an MSTP instance)	32768
Maximum hops	20 hops
Revision number	0

The following table lists the switch defaults for the 10-gigabit Ethernet DCB interface-specific configuration.

TABLE 10 Default 10-gigabit Ethernet DCB interface-specific configuration

Parameter	Default setting
Spanning tree	Disabled 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
Port fast	Disabled
Port priority	128
DCB interface root port	Allow the DCB interface to become a root port.
DCB interface BPDU restriction	Restriction is disabled.

Configuring STP

The process for configuring STP is as follows:

1. Enter global configuration mode.
2. Enable STP by using the global **protocol spanning-tree** command.

```
switch(config)# protocol spanning-tree stp
```

3. Designate the root switch by using the **bridge-priority** command. The range is 0 through 61440 and the priority values can be set only in increments of 4096.

```
switch(conf-stp)# bridge-priority 28672
```

4. Enable port fast on switch ports by using the **spanning-tree portfast** command.

ATTENTION

Note the following conditions:

- Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat these commands for every port connected to workstations or PCs. Do not enable port fast on ports that connect to other switches.
- If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a shut/no shut.
- Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.

```
switch(config)# interface tengigabitethernet 1/0/10
switch(config-if-te-1/0/10)# spanning-tree portfast
switch(config-if-te-1/0/10)# exit
```

5. To interoperate with non-VDX devices (such as NetIron and FastIron) in PVST+/R-PVST+ mode, you may need to configure the interface that is connected to that switch by using the **spanning-tree bpdu-mac** command.

```
switch(config)# interface tengigabitethernet 1/0/12
switch(config-if-te-0/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
```

6. Specify port priorities by using the **spanning-tree priority** command to influence the selection of root/designated ports.
7. Return to privileged EXEC mode.

```
switch(config-if-te-1/0/12)# end
```

When the spanning tree topology is completed, the network switches send and receive data only on the ports that are part of the spanning tree. Data received on ports that are not part of the spanning tree is blocked.

NOTE

Extreme recommends leaving other STP variables at their default values.

Configuring RSTP

The process for configuring RSTP is as follows.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enable RSTP by using the global **protocol spanning-tree** command.

```
switch(config)# protocol spanning-tree rstp
```

3. Designate the root switch by using the **bridge-priority** command. The range is 0 through 61440 and the priority values can be set only in increments of 4096.

```
switch(config-stp)# bridge-priority 28582
```

4. Configure the bridge forward delay value.

```
switch(config-stp)# forward-delay 20
```

5. Configure the bridge maximum aging time value.

```
switch(config-stp)# max-age 25
```

6. Enable the error-disable-timeout timer.

```
switch(config-stp)# error-disable-timeout enable
```

7. Configure the error-disable-timeout interval value.

```
switch(config-stp)# error-disable-timeout interval 60
```

8. Configure the port-channel path cost.

```
switch(config-stp)# port-channel path-cost custom
```

9. Configure the bridge hello-time value.

```
switch(config-stp)# hello-time 5
```

10. Enable port fast on switch ports by using the **spanning-tree portfast** command.

NOTE

Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat these commands for every port connected to workstations or PCs. Do not enable port fast on ports that connect to other switches.

NOTE

Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.

```
switch(config)# interface tengigabitethernet 1/0/10
switch(config-if-te-1/0/10)# spanning-tree portfast
```

11. Specify port priorities by using the **spanning-tree priority** command to influence the selection of root/designated ports.

12. Return to privileged EXEC mode.

```
switch(config)# end
```

Configuring MSTP

The process for configuring MSTP is as follows.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enable MSTP by using the global **protocol spanning-tree** command.

```
switch(config)# protocol spanning-tree mstp
```

3. Specify the region name by using the **region** *region_name* command.

```
switch(config-mstp)# region Extreme1
```

4. Specify the revision number by using the **revision** command.

```
switch(config-mstp)# revision 1
```

5. Map a VLAN to an MSTP instance by using the **instance** command.

```
switch(config-mstp)# instance 1 vlan 2, 3
switch(config-mstp)# instance 2 vlan 4-6
switch(config-mstp)# instance 1 priority 4096
```

- Specify the maximum hops for a BPDU to prevent the messages from looping indefinitely on the interface by using the **max-hops** *hop_count* command.

```
switch(config-mstp)# max-hops 25
```

- Return to privileged EXEC mode.

```
switch(config)# end
```

Configuring additional MSTP parameters

The following sections discuss how to configure additional MSTP parameters.

Enabling and disabling Cisco interoperability (MSTP)

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. The default is Cisco interoperability is disabled.

NOTE

The **cisco-interoperability** command is necessary because the "version 3 length" field in the MSTP BPDU on some legacy Cisco switches does not conform to current standards.

To enable interoperability with certain legacy Cisco switches, perform the following steps from privileged EXEC mode.

- Enter the **config** command to change to global configuration mode.

```
switch# config
```

- Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

- Enter the **cisco-interoperability enable** command to enable interoperability with certain legacy Cisco switches.

```
switch(config-mstp)# cisco-interoperability enable
```

- (Optional) To disable interoperability with certain legacy Cisco switches, enter the **cisco-interoperability disable** command.

```
switch(config-mstp)# cisco-interoperability disable
```

Mapping a VLAN to an MSTP instance

In MSTP mode, use the **instance** command to map a VLAN to an MSTP. You can group a set of VLANs to an instance. 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.

To map a VLAN to an MSTP instance, perform the following steps from privileged EXEC mode.

- Enter the **config** command to change to global configuration mode.

```
switch# config
```

- Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Map a VLAN to an MSTP instance.

```
switch(config-mstp)# instance 5 vlan 300
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

Specifying the maximum number of hops for a BPDU (MSTP)

In MSTP mode, use the **max-hops** command to configure the maximum number of hops for a BPDU in an MSTP region. Specifying the maximum hops for a BPDU prevents the messages from looping indefinitely on the interface. When you change the number of hops, it affects all spanning tree instances. The range is 1 through 40. The default is 20 hops.

To configure the maximum number of hops for a BPDU in an MSTP region, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **max-hops 30** command to change the maximum number of hops from the default for a BPDU in an MSTP region.

```
switch(config-mstp)# max-hops 30
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

Specifying a name for an MSTP region

In MSTP mode, use the **region** command to assign a name to an MSTP region. The region name has a maximum length of 32 characters and is case-sensitive.

To assign a name to an MSTP region, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.

```
switch# config
```

2. Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **region** command to assign a name to an MSTP region.

```
switch(config-mstp)# region sydney
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```


Specifying a revision number for MSTP configuration

In MSTP mode, use the **revision** command to specify a revision number for an MSTP configuration. The range is 0 through 255. The default is 0.

To specify a revision number for an MSTP configuration, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree mstp** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **revision** command to specify a revision number for an MSTP configuration.

```
switch(config-mstp)# revision 17
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

Configuring PVST+ or R-PVST+

To configure PVST+ or R-PVST+, use the **protocol spanning-tree pvst** and **protocol spanning-tree rpvt** commands. Refer to the *Extreme Network OS Command Reference* for details.

The following example script configures PVST+:

```
switch(config)# protocol spanning-tree pvst
switch(conf-pvst)# bridge-priority 4096
switch(conf-pvst)# forward-delay 4
switch(conf-pvst)# hello-time 2
switch(conf-pvst)# max-age 7
```

The following example script configures R-PVST+:

```
switch(config)# protocol spanning-tree rpvt
switch(conf-pvst)# bridge-priority 4096
switch(conf-pvst)# forward-delay 4
switch(conf-pvst)# hello-time 2
switch(conf-pvst)# max-age 7
```

Enabling STP, RSTP, MSTP, PVST+ or R-PVST+

Enable STP to detect and avoid loops. By default, STP, RSTP, MSTP, PVST+, or R-PVST+ are not enabled. You must turn off one form of STP before turning on another form.

Perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** *mode_name* command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree pvst
```

To disable STP, RSTP, MSTP, PVST+, or R-PVST+, enter the **no protocol spanning-tree** command:

```
switch(config)# no protocol spanning-tree
```

NOTE

The above command deletes the context and all the configurations defined within the context or protocol for an interface.

Shutting down STP, RSTP, MSTP, PVST+, or R-PVST+ globally

To shut down STP, RSTP, MSTP, PVST+, or R-PVST+ globally, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **config** command to change to any of the STP configuration modes. This example uses MSTP mode.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **shutdown** command to shut down STP, RSTP, MSTP, PVST+, or R-PVST+ globally. The **shutdown** command works in any STP modes.

```
switch(config-mstp)# shutdown
```

Specifying bridge parameters

There are a variety of options for configuring the behavior of the STP bridging function, as shown in the following subsections.

Specifying the bridge priority

In PVST+ or R-PVST+ mode, use the **bridge-priority** command to specify the priority of the switch. After you decide on the root switch, set the appropriate values to designate the switch as the root switch. If a switch has a bridge priority that is lower than that of all the other switches, the other switches automatically select the switch as the root switch.

The root switch should be centrally located and not in a "disruptive" location. Backbone switches typically serve as the root switch because they often 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 switch.

Bridge Protocol Data Units (BPDUs) carry the information exchanged between switches. When all the switches in the network are powered up, they start the process of selecting the root switch. Each switch transmits a BPDU to directly connected switches on a per-VLAN basis. Each switch compares the received BPDU to the BPDU that the switch sent. In the root switch selection process, if switch 1 advertises a root ID that is a lower number than the root ID that switch 2 advertises, switch 2 stops the advertisement of its root ID, and accepts the root ID of switch 1. The switch with the lowest bridge priority becomes the root switch.

Additionally, you may 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.

To specify the bridge priority, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable PVST+ or R-PVST+.

```
switch(config)# protocol spanning-tree pvst
```

3. Specify the bridge priority. The range is 0 through 61440 and the priority values can be set only in increments of 4096. The default priority is 32678.

```
switch(conf-stp)# bridge-priority 20480
```

4. Specify the bridge priority for a specific VLAN.

```
switch(conf-stp)# bridge-priority 20480 vlan 10
```

Specifying the bridge forward delay

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **forward-delay** command to specify 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. The following relationship should be kept:

$$2 * (\text{forward_delay} - 1) \geq \text{max_age} \geq 2 * (\text{hello_time} + 1)$$

Additionally, you may specify the forward delay 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.

To specify the bridge forward delay, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the bridge forward delay.

```
switch(conf-stp)# forward-delay 20
```

4. Specify the bridge forward delay for a specific VLAN.

```
switch(conf-stp)# forward-delay 20 vlan 10
```

Specifying the bridge maximum aging time

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **max-age** command to control the maximum length of time that passes before an interface saves its BPDU configuration information.

When configuring the maximum aging time, you must set the max-age to be greater than the hello time. The range is 6 through 40 seconds. The default is 20 seconds. The following relationship should be kept:

$$2 * (\text{forward_delay} - 1) \geq \text{max_age} \geq 2 * (\text{hello_time} + 1)$$

Additionally, 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.

To specify the bridge maximum aging time, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the bridge maximum aging time.

```
switch(conf-stp)# max-age 25
```

4. (Optional) Specify the bridge maximum aging time for a specific VLAN.

```
switch(conf-stp)# max-age 25 vlan 10
```

Specifying the bridge hello time

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **hello-time** command to configure the bridge hello time. The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The range is from 1 through 10 seconds. The default is 2 seconds.

When configuring the hello time, you must set the maximum age to be greater than the hello time. The following relationship should be kept:

$$2 * (\text{forward_delay} - 1) \geq \text{max_age} \geq 2 * (\text{hello_time} + 1)$$

Additionally, you may 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.

To specify the bridge hello time, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# configure terminal
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the time range in seconds for the interval between the hello BPDUs sent on an interface.

```
switch(conf-stp)# hello-time 5
```

4. (Optional) Specify the time range in seconds for the interval between the hello BPDUs sent on an interface for a specific VLAN.

```
switch(conf-stp)# hello-time 5 vlan 10
```

5. Return to privileged EXEC mode.

```
switch(config)# end
```

Configuring STP timers

You must configure the error disable timeout timer before you can use it.

Enabling the error disable timeout timer

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **error-disable-timeout** command to enable a timer that will bring a port out of the disabled state. When the STP BPDU guard disables a port, the port remains in the disabled state unless the port is enabled manually. This command allows you to enable the port from the disabled state. For details on configuring the error disable timeout interval, refer to [Specifying the error disable timeout interval](#) on page 101.

To enable the error disable timeout timer, perform the following steps from privileged EXEC mode. By default, the timeout feature is disabled.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Enable the error disable timeout timer.

```
switch(conf-stp)# error-disable-timeout enable
```

Specifying the error disable timeout interval

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **error-disable-timeout** command to 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.

To specify the time in seconds it takes for an interface to time out, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the time in seconds it takes for an interface to time out.

```
switch(conf-stp)# error-disable-timeout interval 60
```

Specifying the port-channel path cost

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **port-channel path-cost** command to specify the port-channel path cost. The default port cost is **standard**. The path cost options are as follows:

- **custom** — Specifies that the path cost changes according to the port-channel's bandwidth.
- **standard** — Specifies that the path cost does not change according to the port-channel's bandwidth.

To specify the port-channel path cost, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the port-channel path cost.

```
switch(config-stp)# port-channel path-cost custom
```

4. Return to privileged EXEC mode.

```
switch(config)# end
```

Specifying the transmit hold count (RSTP, MSTP, and R-PVST+)

In RSTP, MSTP, or R-PVST+ mode, use the **transmit-holdcount** command to configure the BPDU burst size by specifying the transmit hold count value. The command configures the maximum number of BPDUs transmitted per second for RSTP and MSTP before pausing for 1 second. The range is 1 through 10. The default is 6.

To specify the transmit hold count, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Specify the transmit hold count.

```
switch(config-mstp)# transmit-holdcount 5
```

3. Return to privileged EXEC mode.

```
switch(config)# end
```

Clearing spanning tree counters

To clear spanning tree counters on all interfaces or on the specified interface, use the **clear spanning-tree counter** command in privileged EXEC mode.

1. Use the **clear spanning-tree counter** command to clear the spanning tree counters on all the interfaces.

```
switch# clear spanning-tree counter
```

2. Alternatively, use the **clear spanning-tree counter interface** command to clear the spanning tree counters associated with a specific port-channel or DCB port interface.

```
switch# clear spanning-tree counter interface tengigabitethernet 1/0/1
```

Clearing spanning tree-detected protocols

To restart the protocol migration process (force the renegotiation with neighboring switches) on either all interfaces or on a specified interface, use the **clear spanning-tree detected-protocols** command in privileged EXEC mode.

To restart the protocol migration process, perform either of the following tasks:

1. Use the **clear spanning-tree detected-protocols** command to clear all spanning tree counters on all interfaces:

```
switch# clear spanning-tree detected-protocols
```

2. Alternatively, use the **clear spanning-tree detected-protocols interface** *interface_ID* command to clear the spanning tree counters associated with a specific port-channel or DCB port interface:

```
switch# clear spanning-tree detected-protocols interface tengigabitethernet 0/1
```

Displaying STP, RSTP, MSTP, PVST+, or R-PVST+ information

Enter the **show spanning-tree brief** command in privileged EXEC mode to display all STP, RSTP, MSTP, PVST+, or R-PVST+-related information.

NOTE

The **show spanning-tree brief** command output shows the port state as ERR, not root_inc, when **root guard** is in effect.

Configuring STP, RSTP, or MSTP on DCB interface ports

By default, STP is not enabled on individual ports. This section details the commands for enabling and configuring STP, RSTP, or MSTP on individual 10-gigabit Ethernet Data Center Bridging (DCB) interface ports.

Enabling and disabling STP (DCB)

Do the following to enable or disable spanning tree, as appropriate.

Enabling spanning tree (DCB)

From the DCB interface, use this command to enable spanning tree on the DCB interface. By default, spanning tree is disabled.

To enable spanning tree on the DCB interface, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **no spanning-tree shutdown** command to enable spanning tree on the DCB interface.

```
switch(conf-if-te-1/0/1)# no spanning-tree shutdown
```

Disabling spanning tree (DCB)

From the DCB interface, use the **spanning-tree shutdown** command to disable spanning tree on the DCB interface. By default, spanning tree is disabled.

To disable spanning tree on the DCB interface, perform the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree shutdown** command to disable spanning tree on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree shutdown
```

Enabling automatic edge detection (DCB)

From the DCB interface, use the **spanning-tree autoedge** command to identify the edge port automatically. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

To enable automatic edge detection on the DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree autoedge** command to enable automatic edge detection on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree autoedge
```

Configuring the path cost (DCB)

From the DCB interface, use the **spanning-tree cost** command to configure the path cost for spanning tree calculations. The lower the path cost means there is a greater chance of the interface becoming the root port. The range is 1 through 200000000. The default path cost is 2000 for a 10-gigabit Ethernet interface.

Additionally, you may specify the spanning tree cost 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.

To configure the path cost for spanning tree calculations on the DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```


3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree cost** command to configure the path cost for spanning tree calculations on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree cost 10000
```

5. Enter the **spanning-tree vlan** command to configure the path cost for spanning tree calculations on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree vlan 10 cost 10000
```

6. Return to privileged EXEC mode.

```
switch(conf-if-te-1/0/1)# end
```

Enabling a port (interface) as an edge port (DCB)

From the DCB interface, use the **spanning-tree edgeport** command to enable the port as an edge port to allow the port to quickly transition to the forwarding state.

NOTE

The **spanning-tree edgeport** command is only for RSTP and MSTP. Use the **spanning-tree portfast** command for STP (refer to [Enabling port fast \(DCB\)](#) on page 107).

Follow these guidelines to configure a port as an edge port:

- 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.

To enable the DCB interface as an edge port, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree edgeport** command to enable the DCB interface as an edge port.

```
switch(conf-if-te-1/0/1)# spanning-tree edgeport
```

Enabling guard root (DCB)

From the DCB interface, use this command to enable the guard root feature on the switch. This feature provides a way to enforce the root bridge placement in the network. With guard root enabled on an interface, the switch is able to restrict which interface is allowed to be the spanning tree root port or the path to the root for the switch. The root port provides the best path from the switch to the root switch. By default, guard root is disabled.

Guard root 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. This causes severe bottlenecks in the data path. Guard root ensures that the port on which it is enabled is a designated port. If the guard-root-enabled port receives a superior BPDU, it goes to a discarding state.

Additionally, you may enable guard root 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.

To enable guard root on a DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree guard root** command to enable guard root on a DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree guard root
```

5. Optionally, enter the **spanning-tree** command to enable guard root for a specific VLAN.

```
switch(conf-if-te-1/0/1)# spanning-tree guard root vlan 10
```

Specifying the STP hello time (DCB)

From the DCB interface, use **spanning-tree hello-time** command to set the time interval between BPDUs sent by the root switch. Changing the hello time affects all spanning tree instances.

The maximum age setting must be greater than the hello time setting (refer to [Specifying the bridge maximum aging time](#) on page 99). The range for the **spanning-tree hello-time** command is 1 through 10 seconds. The default value is 2 seconds.

To specify the MSTP hello time on a DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree** command to specify the hello time on a DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree hello-time 5
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-1/0/1)# end
```

Specifying restrictions for an MSTP instance (DCB)

From the DCB interface, use the **spanning-tree instance** command to specify restrictions on the interface for an MSTP instance.

To specify restrictions for an MSTP instance on a DCB interface, perform the following steps.

1. Enter the **config** command to access global configuration mode from privileged EXEC mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

The **gigabitethernet** *rbridge-id/slot/port* operand is used only for the Extreme VDX 6710, Extreme VDX 8770-4, and Extreme VDX 8770-8. The prompt for these ports is in the following example format: `switch(config-if-gi-22/0/1)#`

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(config-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree instance restricted-tcn** command to restrict Topology Change Notification (TCN) BPDUs for an MSTP instance on a DCB interface.

```
switch(config-if-te-1/0/1)# spanning-tree instance 5 restricted-tcn
```

5. Return to privileged EXEC mode.

```
switch(config-if-te-1/0/1)# end
```

Specifying a link type (DCB)

From the DCB interface, use the **spanning-tree link-type** command to specify a link type. Specifying the **point-to-point** keyword enables rapid spanning tree transitions to the forwarding state. Specifying the **shared** keyword disables spanning tree rapid transitions. The default setting is point-to-point.

To specify a link type on a DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(config-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree link-type shared** command, as in the following example, to change the link type from the default.

```
switch(config-if-te-1/0/1)# spanning-tree link-type shared
```

Enabling port fast (DCB)

From the DCB interface, use the **spanning-tree portfast** command to enable port fast on an interface to allow the interface to transition quickly to the forwarding state. Port fast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

NOTE

If you enable the **portfast bpduguard** option on an interface and the interface receives a BPDU, the software disables the interface and puts the interface in the ERR_DISABLE state.

**CAUTION**

Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.

Use the **spanning-tree edgeport** command for MSTP, RSTP, and R-PVST+ (refer to [Enabling a port \(interface\) as an edge port \(DCB\)](#) on page 105).

To enable port fast on the DCB interface for STP, perform the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-0/1)# no shutdown
```

4. Enter the **spanning-tree portfast** command to enable port fast on the DCB interface.

```
switch(conf-if-te-0/1)# spanning-tree portfast
```

Specifying the port priority (DCB)

From the DCB interface, use the **spanning-tree priority** command to specify the port priority. The range is from 0 through 240 in increments of 16. The default value is 128.

In addition, you may specify the spanning tree 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.

To specify the port priority on the DCB interface, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree** command to specify the port priority on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree priority 32
```

5. (Optional) Enter the **spanning-tree vlan priority** command to specify the port priority for a specific VLAN.

```
switch(conf-if-te-1/0/1)# spanning-tree vlan 10 priority 32
```

Restricting the port from becoming a root port (DCB)

From the DCB interface, use the **spanning-tree restricted-role** command to restrict a port from becoming a root port. The default is to allow the DCB interface to become a root port. This procedure affects MSTP only.

To restrict the DCB interface from becoming a root port, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.

2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree restricted-role** command to restrict the DCB interface from becoming a root port.

```
switch(conf-if-te-1/0/1)# spanning-tree restricted-role
```

Restricting the topology change notification (DCB)

From the DCB interface, use the **spanning-tree restricted-tcn** command to restrict the Topology Change Notification (TCN) BPDUs sent on the interface. By default, the restriction is disabled. This procedure affects MSTP only.

To restrict the TCN BPDUs sent on the DCB interface, perform the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree restricted-tcn** command to restrict the TCN BPDUs sent on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree restricted-tcn
```

Configuring DiST

By default, spanning tree is disabled at the global and interface configuration levels. With respect to Distributed STP (DiST), server ports must be configured as an xSTP edge port.

NOTE

DiST is supported on the VCS edge ports only. DiST cannot be enabled on ISL ports participating in the TRILL-based fabric within VCS. DiST does not update the port state of ISL ports.

xSTP can be enabled on the VCS by means of the **protocol spanning-tree** command. An interface begins participating in the spanning tree once it is configured by the **spanning-tree enable** command. Refer to [Configuring and managing STP and STP variants](#) on page 91.

The following table describes the behavior of interface based on global and interface level configuration.

TABLE 11 Interface behavior by global and interface configuration

Global config	Interface config	Interface type where STP BPDU is received	Action
Disable	N/A	Layer 2 (switchport, FCoE)	Flood to all Layer 2 ports
Disable	N/A	Layer 3	Drop
Enable	Disable	Layer 2 (switchport, FCoE)	Drop
Enable	N/A	Layer 3	Drop

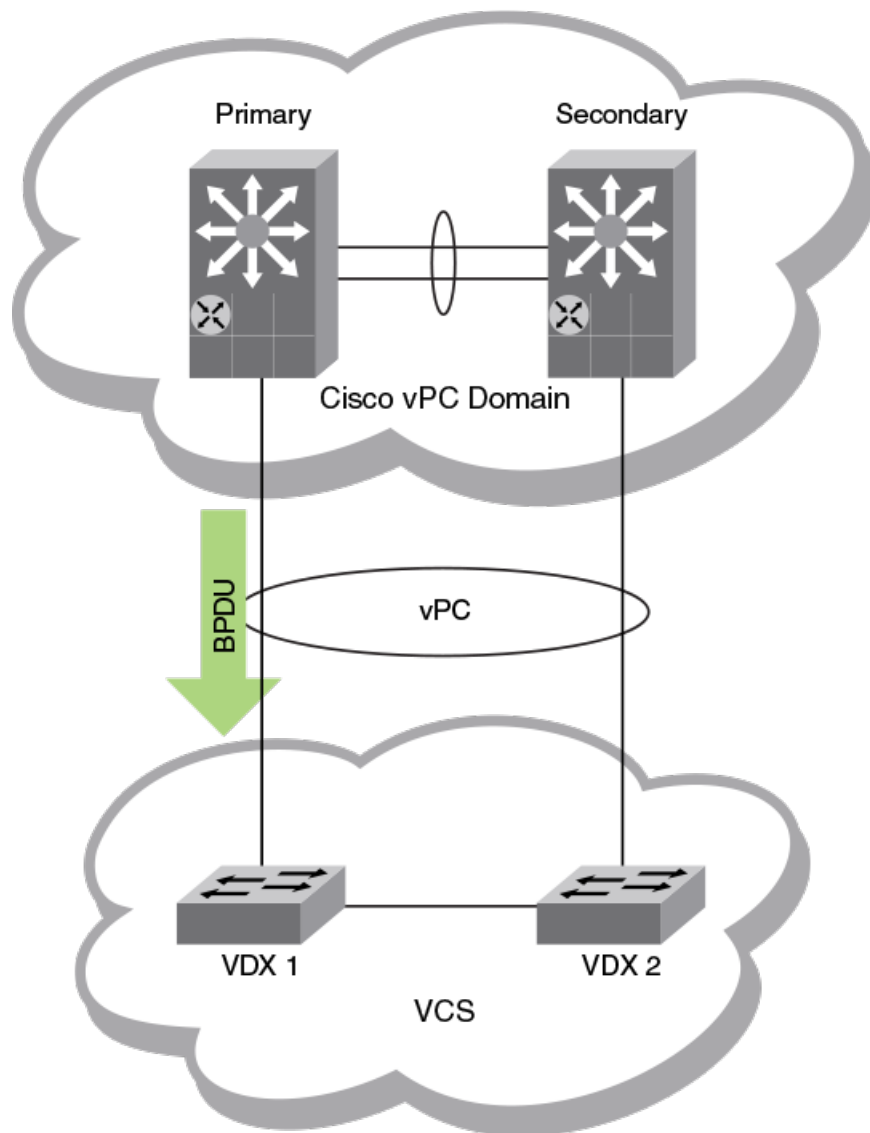
TABLE 11 Interface behavior by global and interface configuration (continued)

Global config	Interface config	Interface type where STP BPDU is received	Action
Enable	Enable	(switchport, FCoE)	Trap

Cisco Peer-Switch support

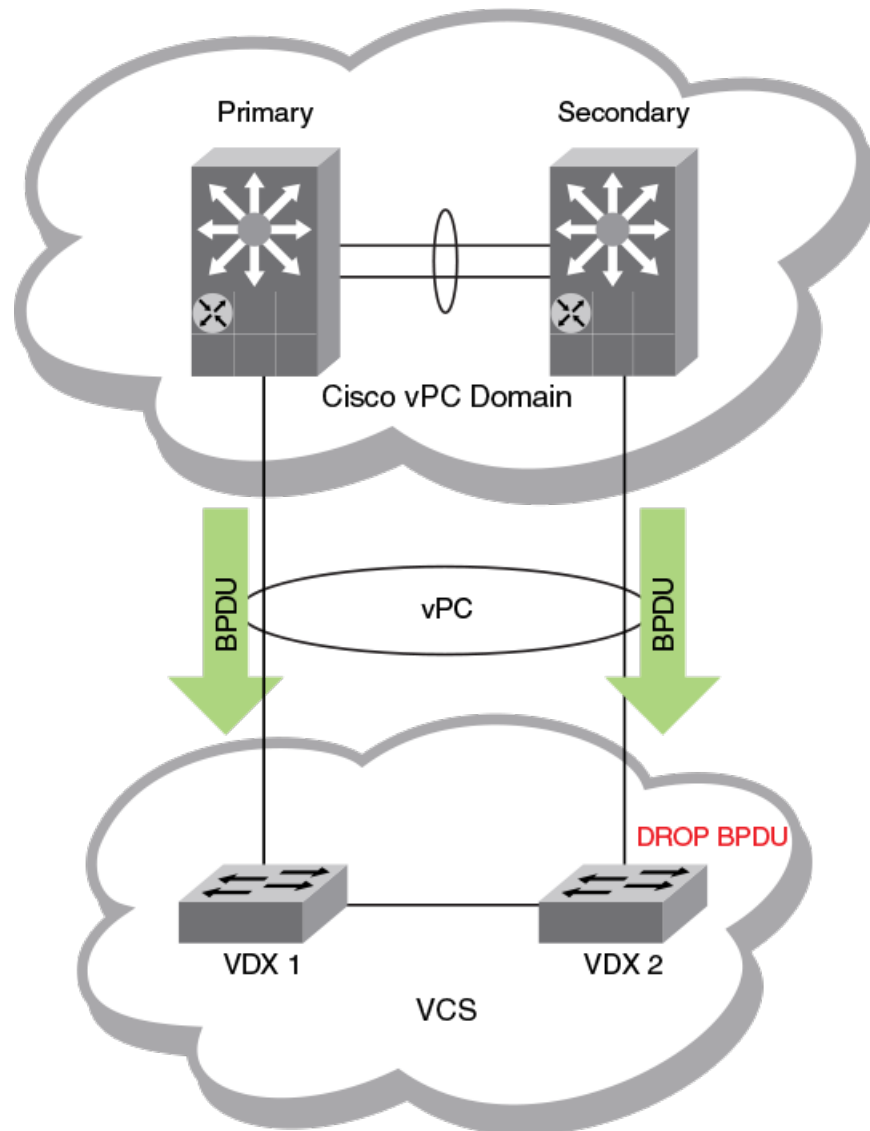
When the Peer-Switch feature is enabled on a Cisco vPC domain, it broadcasts the same BPDUs from both vPC primary and secondary nodes to peer devices. However, a VCS on a VLAG assumes that any logical interface receives only one BPDU from any of its member ports. Consequently, when the VCS receives the two BPDUs from a Cisco vPC domain, it creates a churn of VLAG mastership that increases the CPU load on an Extreme VDX. To avoid this problem, BPDUs received on the VLAG non-master are dropped when the Peer-Switch feature is enabled on the VLAG. The following figure illustrates STP BPDU processing without the Peer-Switch feature.

FIGURE 16 STP BPDU processing without Peer-Switch



In the following figure, where the Peer-Switch feature is enabled, the Extreme VDX 1 receives the BPDU, so it becomes the VLAG master and VDX 2 is set in the non-master state. VDX 1 remains the VLAG master as long as it receives BPDUs.

FIGURE 17 STP BPDU Processing with Peer-Switch



When the Peer-Switch functionality is enabled and the VLAG Master is selected, BPDUs received on VLAG non-master are dropped unless there is a change in the status of the VLAG master. By default, the Peer-Switch feature functionality is inactive. To activate this function, refer to the **spanning-tree peer-switch** command in the *Network OS Command Reference*.

NOTE

The Peer-Switch feature works only when MSTP is enabled on Cisco switches and an Extreme VDX. It does not work with other flavors of STP. In addition, the Peer-Switch feature is not supported for PVST/RPVST mode unless a Cisco device sends the same BPDU from both the primary and secondary vPC nodes. Currently it sends two different BPDUs. Cisco documentation says that the same BPDU is sent from both primary and secondary nodes of a vPC domain; however, when RPVST is enabled then a message age variable in the BPDU sent from a secondary node is different from that sent from a primary node.

UDLD

- [UDLD overview.....](#) 113
- [Configuring UDLD.....](#) 114
- [Additional UDLD-related commands.....](#) 115

UDLD overview

The UniDirectional Link Detection (UDLD) protocol is a nonstandard Layer 2 protocol that detects when a physical link becomes unidirectional by means of the exchange of UDLD protocol data units (PDUs). A unidirectional loop can lead to the creation of a loop in a network, which the Spanning Tree Protocol (STP) could inadvertently allow to occur.

UDLD requirements

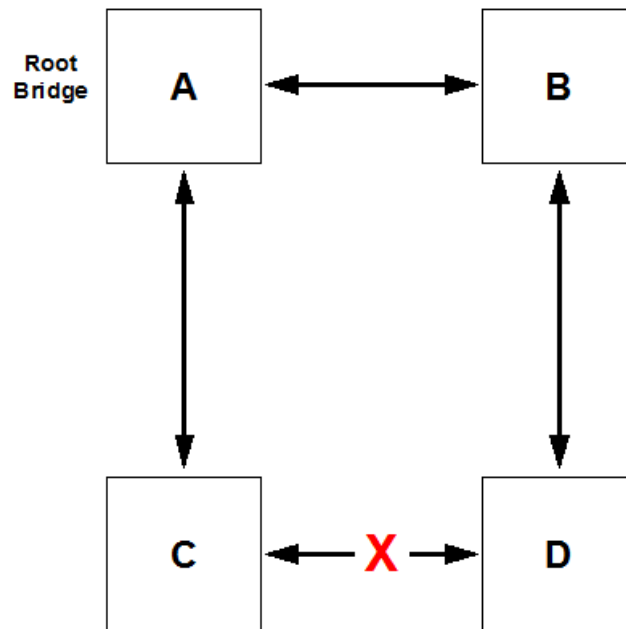
Note the following requirements for UniDirectional Link Detection:

- Network OS 4.0 or later.
- UDLD runs only on physical ports assigned to a port channel.
- UDLD is supported on directly connected switches only.
- In VCS mode, UDLD applies only to edge ports.
- UDLD can interoperate with Extreme IP products.

How UDLD works

The following shows a simple four-switch network in which two paths connect to each switch. STP blocks traffic on as many ports as necessary so that only one operational path exists from the STP root bridge to all nodes in the network.

FIGURE 18 Four-switch example for UDLD



In the figure above, STP detects that the port on switch D that is connected to switch C should be put into a blocked state. Therefore, no data traffic gets transmitted or received on this port. Data traffic remains blocked as long as switch D receives bridge protocol data units (BPDUs) from both switches C and B.

If the link between switch C and switch D becomes unidirectional (for reasons such as hardware failure or incorrect cabling) in the direction from D to C, switch D ages out the status that it was receiving BPDUs from switch C. This eventually causes STP to put the port in a forwarding state, thus allowing all data traffic. This creates a loop for all BUM traffic that enters the network. BUM traffic can go from switch B to switch D to switch C to switch A, and then back to switch B.

To prevent this loop from forming, UDLD can be used to detect that the link between switch C and switch D has become unidirectional.

The UDLD protocol is disabled by default. To use the UDLD protocol, you must first enable the protocol globally and then enable UDLD on each desired individual physical port. For a configuration example, refer to [UDLD](#) on page 113.

UDLD determines that a link has become unidirectional if the UDLD daemon stops receiving UDLD PDUs from the other end of the link. The UDLD daemon then blocks the physical link. The physical link remains up but the line protocol goes down. During this time, the link continues to transmit and receive UDLD PDUs.

NOTE

In a VCS environment, the UDLD protocol is applicable only to the edge ports in the VCS. A configuration command to enable the UDLD protocol on a logical port or a non-edge port will be rejected.

Configuring UDLD

Follow the steps below to configure basic UDLD on your switch.

1. Enter global configuration mode by entering the **configure** command from the desired switch:

```
switch# configure
```

- To enable the UDLD protocol, as well as to enter protocol UDLD configuration mode, enter the **protocol udld** command.

```
switch(config)# protocol udld
```

- (Optional) You can change the interval at which UDLD PDUs are transmitted from edge ports. The default interval, in counts of one hundred milliseconds, is 5 (500 milliseconds). To change the interval to 2,000 milliseconds, enter the **hello 20** command:

```
switch(config-udld)# hello 20
```

- You can change the timeout multiplier value to affect the UDLD PDU timeout interval. The UDLD timeout interval is the product of the hello time interval at the other end of the link and the timeout multiplier value. To change the timeout multiplier from the default of 5 to the value 8, run the **multiplier 8** command:

```
switch(config-udld)# multiplier 8
```

- Enter interface subconfiguration mode for the edge port on which you want to enable UDLD:

```
switch(config-udld)# end
switch# configure
switch(config)# interface te 5/0/1
switch(config-int-te-5/0/1)# udld enable
```

- Repeat the preceding step for each edge port on which you wish to enable UDLD.

NOTE

When the UDLD protocol is enabled on one end of a link, the timeout period might elapse before the UDLD protocol is enabled on the other end of the link. In this case, the link becomes temporarily blocked. When the UDLD protocol is enabled at the other end of the link and a UDLD PDU is received, UDLD automatically unblocks the link.

Additional UDLD-related commands

Among additional UDLD commands that you can use are the following:

- clear udld statistics** — Clears either all UDLD statistics or clears the statistics on a specified port.
- debug udld packet** — Enables debugging for UDLD
- show debug udld** — Displays UDLD debug status on the switch.
- show udld** — Displays global UDLD information.
- show udld interface** — Displays UDLD information for one or all ports.
- show udld statistics** — Displays either all UDLD statistics or the statistics on a specified port.

For more information about how to use UDLD commands, refer to the *Extreme Network OS Command Reference*.

Link Aggregation

- [Link aggregation overview.....](#) 117
- [Link aggregation setup.....](#) 124

Link aggregation overview

Link aggregation allows you to bundle multiple physical Ethernet links to form a single logical trunk providing enhanced performance and redundancy. The aggregated trunk is referred to as a Link Aggregation Group (LAG). 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 summarized 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.
- 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 VDX family of switches supports the following trunk types:

- Static, standards-based LAG
- Dynamic, standards-based LAG using LACP
- Static, Extreme-proprietary LAG
- Dynamic, Extreme-proprietary LAG using proprietary enhancements to LACP

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.

Dynamic link aggregation

Dynamic link aggregation uses LACP to negotiate which links 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.

Static link aggregation

In static link aggregation, links are added into a LAG without exchanging LACPDUs 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.

LACP configuration guidelines and restrictions

This section applies to standards-based and Extreme-proprietary LAG configurations, except where specifically noted otherwise.

Follow these LACP configuration guidelines and restrictions when configuring LACP:

- All ports on the Extreme VDX hardware can operate only in full-duplex mode.
- On Extreme-proprietary LAGs only, all LAG member links must be part of the same port-group.
- Interfaces configured as "switchport" interfaces cannot be aggregated into a LAG. However, a LAG can be configured as a switchport.

Extreme-proprietary aggregation

Extreme-proprietary aggregation is similar to standards-based link aggregation but differs in how the traffic is distributed. It also has additional rules that member links must meet before they are aggregated:

- The most important rule requires that there is not a significant difference in the length of the fiber between the member links, and that all member links are part of the same port-group.
- A maximum of four LAGs can be created per port-group.

LAG distribution process and conditions

The LAG aggregator is associated with the collection and distribution of Ethernet frames. The collection and distribution process is required to guarantee the following:

- 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:

- Interfaces configured as switchport interfaces cannot be aggregated into a LAG. However, a LAG can be configured as a switchport.

Virtual LAGs

Configuring a virtual LAG (vLAG) is similar to configuring a LAG. Once the VCS Fabric detects that the LAG configuration spans multiple switches, the LAG automatically becomes a vLAG.

LACP on the VCS Fabric emulates a single logical switch by sending the same LACP system ID and sending the same admin and operational key.

Note these vLAG features :

- Only ports with the same speed are aggregated.
- Proprietary LAGs are not available for vLAGs.
- LACP automatically negotiates and forms the vLAG.
- A port-channel interface is created on all the vLAG members.
- The VCS Fabric relies on you to consistently configure all nodes in the vLAG.
- Similar to static LAGs, vLAGs are not able to detect configuration errors.
- A zero port vLAG is allowed.
- IGMP snooping fits into the primary link of a vLAG to carry multicast traffic.
- Interface statistics are collected and shown per vLAG member switch. The statistics are not aggregated across switches participating in a vLAG.
- In order to provide link and node level redundancy, the VCS Fabric supports static vLAGs.
- A vLAG can be configured within a virtual fabric. For more information about virtual fabrics, refer to the "Virtual Fabrics" chapter.
- A VCS Fabric vLAG functions with servers that do not implement LACP because it supports static vLAGs as well.

vLAG scalability

Dynamic virtual LAGs (vLAGs) are made scalable to support a resilient network infrastructure. To achieve scalability of vLAGs, the system must disable the following vLAG operations at the VCS Fabric level:

- vLAG member link-state update
- Primary RBridge selection
- Actor system ID (SID) selection
- Partner system ID (SID) validation

Use the **vlag-commit-mode disable** command to disable the vLAG operations at the VCS Fabric level and to support vLAG scalability. Use the **show running-config vlag-commit-mode** command to view the current vLAG commit-mode state.

The following system error message is generated when multiple partners are detected for a vLAG. This message is displayed when the **vlag-commit-mode disable** command is configured.

```
2015/01/04-16:33:23, [LACP-1001], 783124, SW/0 | Active | DCE, ERROR, sw0, vLAG multiple partner
detected on Po 110
```

If this system error message appears because of a potential multi-partner vLAG conflict, the user should issue the **show vlag-partner-info** command to verify the multi-partner vLAG configuration and resolve any issues, as shown in the following example.

```
device# show vlag-partner-info
Port-channel 110
RBridge 1: Partner System ID - 0xffff,00-05-33-48-71-a8 Key 0009
RBridge 4: Partner System ID - 0xffff,11-22-33-44-55-66 Key 0009
RBridge 5: Partner System ID - 0xffff,00-05-33-48-71-a8 Key 0009
```

NOTE

The **vlag-commit-mode disable** command cannot be configured if the **no vlag ignore-split** command is configured on any port-channel in the VCS Fabric. Similarly, the **no vlag ignore-split** command cannot be configured if the **vlag commit-mode disable** command is configured.

IP over port-channel

Beginning with Network OS 7.0.0 and the introduction of IP Fabrics, support is provided over port-channels (Layer 2) for Layer 3 protocols.

Support for IP over port-channel provides the following advantages:

- Increases bandwidth between two RBridges.
- Provides fault tolerance, so that there is zero loss until the last member of the port-channel remains.
- Provides for dynamic bandwidth, through the removal or addition of port-channel members. (The upper layers do not need to know about the members, as these are device-dependent.)
- Provides load balancing.

This feature provides support for the following:

- Standard and Network OS port-channels, both static and dynamic
- IPv4 and IPv6 addressing
- Configuration and show commands as are currently supported for physical ports
- High availability

Support is provided for the following Layer 3 protocols:

- ARP/ND
- BFD
- BGP
- DHCP
- ICMP
- IGMP
- OSPFv2/v3
- PIM

- VRRP

In addition, support is provided for route-map policy.

Limitations

Note the following limitations:

- IP over vLAGs is not supported. If a port-channel with member interfaces from more than one node in logical chassis cluster mode, IPv4/IPv6 configurations are not allowed. If a port-channel (vLAG) with members from two nodes becomes segmented, it is no longer a vLAG operationally and IPv4/IPv6 configurations are not allowed.
- sFlow is not supported for Layer 3 port-channels.
- Layer 3 configurations, including IPv4/IPv6 address configurations, are not allowed on an "empty" port-channel (that is, one that has not yet been configured).
- A port-channel cannot be unconfigured until all Layer 3 configurations are removed from it.

Supported commands

The following commands, organized largely by protocol, support IP over port-channel.

IP routing commands

- `ip route`
- `ipv6 route`

Interface commands

- `clear ipv6 counters interface port-channel`
- `ip address`
- `ip address secondary`
- `ip mtu`
- `ipv6 address`
- `ipv6 address secondary`
- `ipv6 address use-link-local-only`
- `ipv6 address eui-64`
- `ipv6 address eui-64 secondary`
- `ipv6 address link-local`
- `ipv6 address anycast`
- `show ip interface`
- `show ipv6 interface`
- `show ip interface brief`
- `show ipv6 interface brief`
- `show ipv6 counters interface port-channel`
- `show port port-channel`
- `show port-channel`
- `show port-channel-redundancy-group`

ARP/ND commands

- arp interface port-channel
- ip proxy-arp
- ip arp-aging-timeout
- show arp port-channel
- show ipv6 neighbor port-channel

BGP commands

- neighbor

DHCP commands

- ip dhcp relay

ICMP commands

- ip icmp

IGMP commands

- ip igmp immediate-leave
- ip igmp last-member-query-count
- ip igmp last-member-query-interval
- ip igmp query-interval
- ip igmp query-max-response-time
- ip igmp robustness-variable
- ip igmp startup-query-count
- ip igmp startup-query-interval
- ip igmp static-group
- show ip igmp interface port-channel
- show ip igmp groups interface port-channel

OSPFv2 commands

- ip ospf active
- ip ospf area
- ip ospf auth-change-wait-time
- ip ospf authentication-key
- ip ospf bfd
- ip ospf cost
- ip ospf database-filter
- ip ospf dead-interval
- ip ospf hello-interval
- ip ospf md5-authentication
- ip ospf mtu-ignore
- ip ospf network
- ip ospf passive
- ip ospf priority

- ip ospf retransmit-interval
- ip ospf transmit-delay
- clear ip ospf counters port-channel
- show ip ospf interface port-channel
- show ip ospf neighbor port-channel
- show debug ip ospf internal interface port-channel

OSPFv3 commands

- ipv6 ospf active
- ipv6 ospf area
- ipv6 ospf authentication
- ipv6 ospf bfd
- ipv6 ospf cost
- ipv6 ospf dead-interval
- ipv6 ospf hello-interval
- ipv6 ospf instance
- ipv6 ospf mtu-ignore
- ipv6 ospf network
- ipv6 ospf passive
- ipv6 ospf priority
- ipv6 ospf retransmit-interval
- ipv6 ospf suppress-linklsa
- ipv6 ospf transmit-delay
- clear ipv6 ospf counts neighbor interface port-channel
- clear ipv6 ospf neighbor interface port-channel
- show ipv6 ospf interface port-channel
- show ipv6 ospf neighbor interface port-channel

PIM commands

- ip multicast-booundary
- ip pim-sparse
- ip pim dr-priority
- ip pim neighbor-filter
- show ip pim-sparse interface port-channel
- show ip pim neighbor interface port-channel

Route-map policy commands

- ip policy route-map
- ipv6 policy route-map
- match interface port-channel
- show route-map interface port-channel

VRRP commands

- `clear vrrp statistics interface port-channel`
- `clear ipv6 vrrp statistics interface port-channel`
- `debug vrrp packets interface port-channel`
- `debug ipv6 vrrp packets interface port-channel`
- `ipv6 vrrp-group`
- `show ipv6 vrrp interface port-channel`
- `show vrrp interface port-channel`
- `vrrp-group`

Ethernet Segment Identifiers (ESIs) for BGP routing

ESIs are used to identify the links connecting multiple ToR devices to a server in a BGP EVPN environment.

If a server is connected to more than one ToR devices over a vLAG interface, the set of links that attaches the server to these devices is referred to as an Ethernet Segment (ES), which is associated with a globally unique Ethernet Segment Identifier, or ESI. When BGP routing is used to support Ethernet Virtual Private Network (EVPN) deployments, each BGP router advertises an Ethernet Segment Route (ESR) to inform other devices that it is connected to that ES. This allows different BGP routers to discover whether they are connected to the same ES. The default ESI value is 0 (Single Home). (The links that connect single-homed devices to server are not required to have an ESI value associated with them.)

The ESI value can be configured manually by means of the `esi` command in port-channel configuration mode. For details, see "Configuring an ESI on a port-channel for BGP routing" later in this chapter.

NOTE

The ESI value can also be derived automatically by means of the LACP Partner SystemID/Port Key. Because this value is the same when the host is dual-homed, the two peer BGP routers derive identical ESI values for the ES and autodiscovery is enabled.

Link aggregation setup

The following sections discuss how to set up link aggregation.

vLAG configuration overview

Network OS 4.0 and later supports the option of setting the "Allowed Speed" of the port-channel to either 1 Gbps or 10 Gbps. The default is 10 Gbps. If the port-channel is 1 Gbps, then the speed needs to be configured before the port-channel is enabled. Otherwise, the physical links are throttled down because of a speed mismatch. Refer to the *Extreme Network OS Command Reference* for information on the `speed` command.

The following conditions and requirements should be kept in mind when configuring vLAGs:

- FCoE and DCB capabilities are not supported by vLAG. FCoE traffic is treated similarly to normal LAN data traffic.
- Static vLAGs are not supported on internal ports.
- Perform this procedure on all member nodes of a vLAG.

Configuring vLAGs

To configure a vLAG, perform the following steps:

1. Change to global configuration mode.
2. Configure a LAG between two switches within the VCS Fabric.

Refer to [LAG distribution process and conditions](#) on page 118 for more information. Once the VCS Fabric detects that the LAG configuration spans multiple switches, the LAG automatically becomes a vLAG.

3. Enter **interface port-channel ID** on every switch in the vLAG to configure them to treat FCoE MAC addresses as being multi-homed hosts, similar to LAN traffic.

The default configuration is to treat FCoE traffic as non-vLAG traffic.

```
switch(config)# interface port-channel 10
```

4. Enter **end** to return to privileged EXEC mode.

```
switch(config-Port-channel-10)# end
switch#
```

5. Enter the **show port-channel detail** command to verify the port-channel details.

```
switch# show port-channel detail
LACP Aggregator: Po 27
Aggregator type: Standard
Ignore-split is disabled
Actor System ID - 0x8000,00-05-33-6f-18-18
Admin Key: 0027 - Oper Key 0027
Receive link count: 4 - Transmit link count: 4
Individual: 0 - Ready: 1
Partner System ID - 0x8000,00-05-1e-cd-6e-9f
Partner Oper Key 0027
Member ports on rbridge-id 231:
Link: Te 231/0/22 (0xE718160201) sync: 1 *
Link: Te 231/0/23 (0xE718170202) sync: 1
Link: Te 231/0/36 (0xE718240305) sync: 1
Link: Te 231/0/37 (0xE718250306) sync: 1
```

6. Enter the **show port port-channel** command to verify the port-channel interface details.

```
switch# show port port-channel tengigabitethernet 1/0/21
LACP link info: te0/21 -0x18150014
Actor System ID: 0x8000,01-e0-52-00-01-00
Actor System ID Mapped Id: 0
Partner System ID: 0x0001,01-80-c2-00-00-01
Actor priority: 0x8000 (32768)
Admin key: 0x000a (10) Operkey: 0x0000 (0)
Receive machine state : Current
Periodic Transmission machine state : Slow periodic
Muxmachine state : Collecting/Distr
Admin state: ACT:1 TIM:0 AGG:1 SYN:0 COL:0 DIS:0 DEF:1 EXP:0
Operstate: ACT:1 TIM:0 AGG:1 SYN:1 COL:1 DIS:1 DEF:0 EXP:0
Partner operstate: ACT:1 TIM:0 AGG:1 SYN:1 COL:1 DIS:1 DEF:0 EXP:0
Partner oper port: 100
```

Configuring VLAG load balancing

Multicast traffic is forwarded only on the primary member port of a vLAG. This can lead to traffic over-subscription, and vLAG load balancing prevents this from occurring.

Within the VCS, for IGMP and Multicast Listener Discovery (MLD) hosts learned through the vLAG traffic are first forwarded to the RBridge that controls the primary port for that vLAG. That RBridge performs Multicast traffic load balancing if it has more than one local member port for that vLAG. Traffic is forwarded to all member ports of the vLAG.

Multicast vLAG Load Balancing can be activated for multiple VLANs that are running the IGMP or MLD snooping protocol. IGMP or MLD notifies the Multicast Sub-System about all VLANs that require load balancing.

The Multicast Sub-System requests HSL to toggle load balancing according to Layer 2 Multicast Group Identifier (MGID). For any MGID port list change on the event of a IGMP/MLD new member add or leave event, the Multicast Sub-System and MGID updates requests and carries information about vLAG load balancing.

The VLAN must be configured before load balancing can be activated.

1. Change to global configuration mode.

```
device# configure terminal
```

2. Change to VLAN configuration mode.

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

3. Use the **ip igmp snooping vlag-load-balancing** command to activate load balancing for IP. Use the **ipv6 mld snooping vlag-load-balancing** command to activate load balancing for IPv6.

```
device(config-Vlan-10)# ip igmp snooping vlag-load-balancing
```

or

```
device(config-Vlan-10)# ipv6 mld snooping vlag-load-balancing
```

4. Use the **show running-config interface** command to confirm that igmp load balancing is active.

```
device(config-Vlan-10)# do show ip igmp int vlan 10
Interface Vlan 10
  IGMP Snooping enabled
  IGMP Snooping fast-leave disabled
  IGMP Snooping restrict-unknown-multicast disabled
  IGMP Snooping vlag-load-balancing enabled
  IGMP Snooping querier enabled
  IGMP query interval is 125 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1 seconds
  IGMP startup-query interval 31 seconds
  IGMP startup-query count 2
  IGMP last-member query count 2
  IGMP robustness variable 2
  Number of router-ports: 0
```

5. Use the **show running-config interface** command to confirm that mld load balancing is active.

```
device(config-Vlan-10)# do show ipv6 mld int vlan 10
Interface Vlan 10
  MLD Snooping enabled
  MLD Snooping fast-leave disabled
  MLD Snooping restrict-unknown-multicast disabled
  MLD Snooping vlag-load-balancing enabled
  MLD Snooping querier disabled
  MLD Snooping query interval is 125 Seconds
  MLD Snooping max query response time is 10 Seconds
  MLD Snooping Last member query response interval is 1 Seconds
  MLD Snooping last-member query count 2
  MLD Snooping startup-query interval 31 Seconds
  MLD Snooping startup-query count 2
  MLD Snooping robustness variable 2
  Number of router-ports: 0
```

Configuring vLAGs to minimize packet loss

This topic provides background on configuring a vLAG to minimize packet loss.

In scenarios where a vLAG spans more than one node, the **vlag ignore-split** command minimizes the extent of packet loss in the event of one of the nodes in the vLAG going down, and also reduces vLAG failover downtime. The scope of this configuration is per port-channel on LACP-based vLAGS.

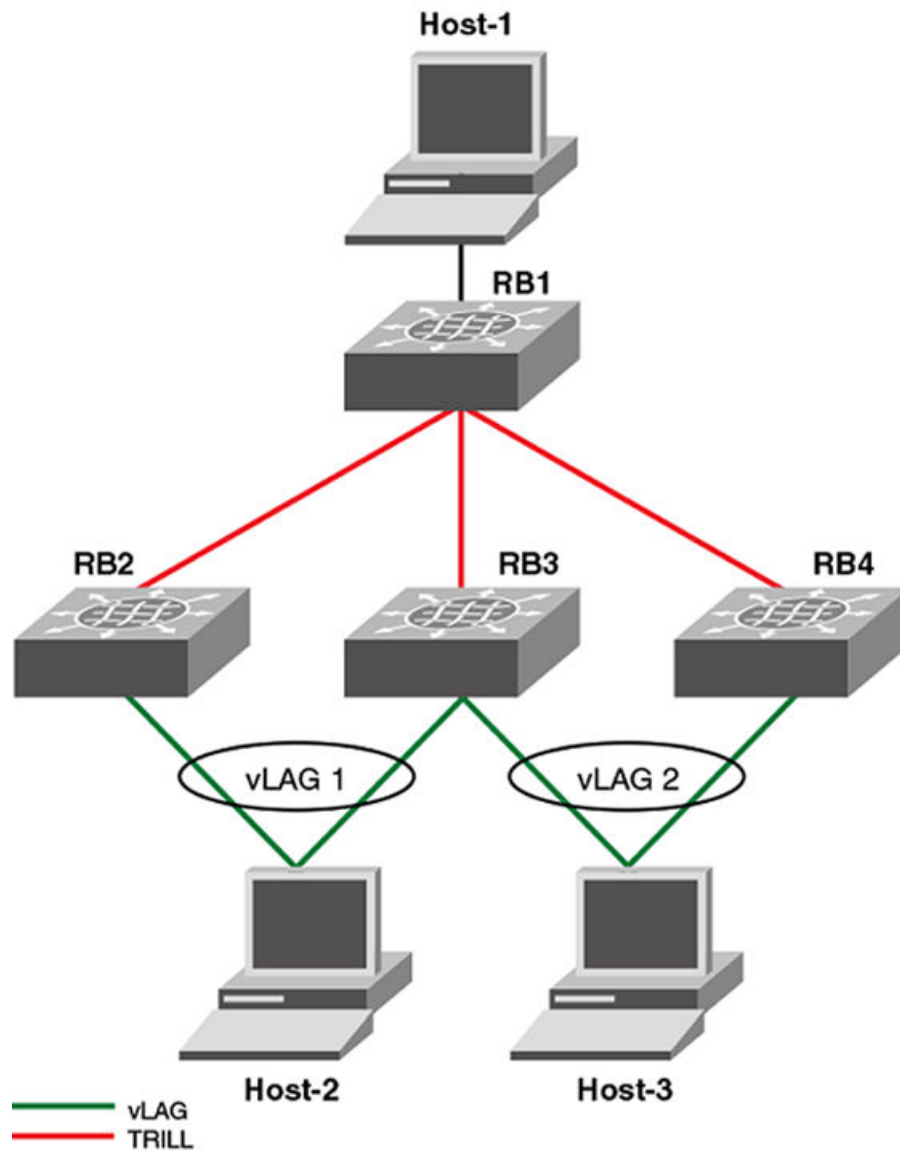
In a case where connectivity between nodes is lost because of a fabric split (as opposed to one of members going down), there will be duplication of multicast/broadcast packets. Extreme recommends that you build redundancy in the fabric so that individual links are not single points of failure.

NOTE

With **ignore-split** active, a vLAG node reboot can result in a more than one-second loss while interoperating with a Linux server/nic-team/CNA, because of premature egress of traffic from the server.

The figure below displays a dual-vLAG configuration with three legs of RB2, RB3, and RB4. If RB2, RB3, or RB4 reboots while Host-1 is communicating to Host-2 or Host3, a momentary traffic disruption may occur.

FIGURE 19 vLAG configuration of the ignore-split feature



To reduce vLAG failover down time, you must configure **ignore-split** on all of the legs in the vLAG (RB2, RB3 and RB4 in this case).

NOTE

By default, **vlag ignore-split** is already activated in VCS.

[Configuring the vLAG ignore-split feature](#) on page 128 walks you through setting up the vLAG ignore-split feature.

Configuring the vLAG ignore-split feature

This topic describes how to configure the vLAG ignore-split feature.

The switch must be in global configuration mode.

To configure the vLAG ignore-split feature, perform the following steps.

NOTE

The following example is based on the illustration in [Configuring vLAGs](#) on page 125.

1. Log in to RB2, the first leg of the vLAG 1.

2. Access the port-channel for the first leg.

```
switch(config)# interface port-channel 1
```

3. Activate vLAG ignore split.

```
switch(config-Port-channel-1)# vlag ignore-split
```

4. Log in to RB3, the second leg of vLAG 1.

5. Access the port-channel for the second leg.

```
switch(config)# interface port-channel 2
```

6. Activate vLAG ignore split.

```
switch(config-Port-channel-2)# vlag ignore-split
```

7. Access the port-channel for the third leg.

```
switch(config)# interface port-channel 3
```

8. Activate vLAG ignore split.

```
switch(config-Port-channel-3)# vlag ignore-split
```

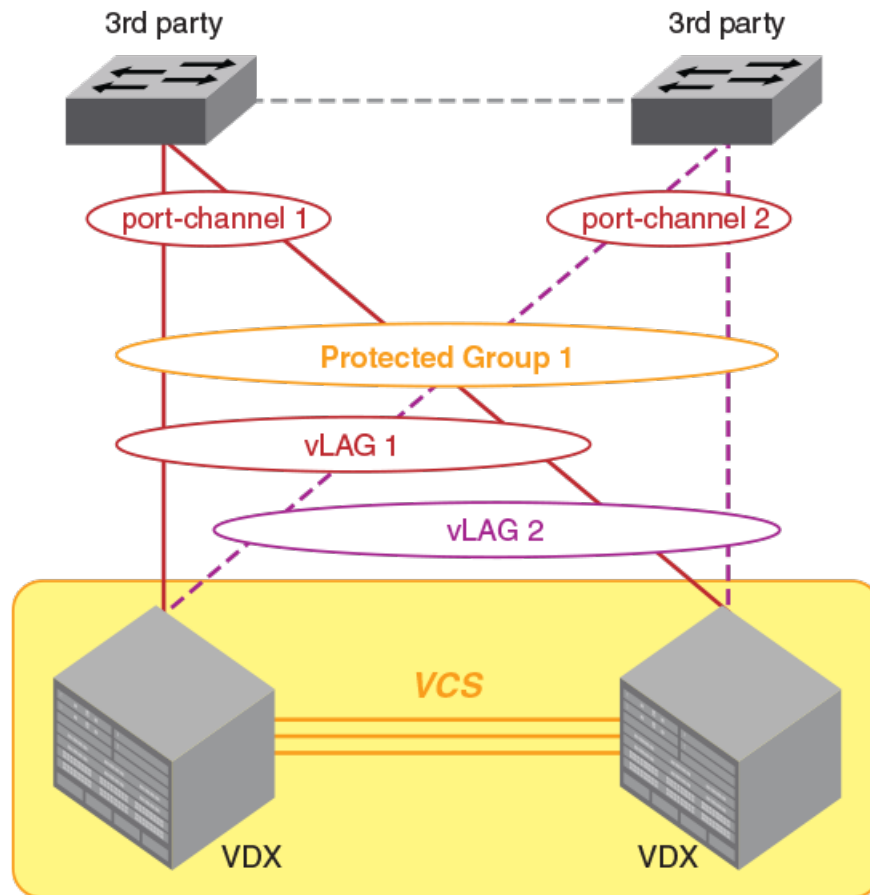
Port-channel redundancy groups

A port-channel redundancy group is used as an alternative to the spanning-tree-protocol for faster convergence and failover in the critical path of the redundant network.

Port-channel redundancy groups minimize disruption to the network by protecting critical links from data loss.

A port-channel redundancy group is a pair vLAGs (or LAGs) configured to act as one active and one backup vLAG connection. The vLAGs and the corresponding port-channels must be configured before the port-channel redundancy groups can be configured. At any point in time only one vLAG is actively forwarding the traffic and the other vLAG is in a discarding state. If the active vLAG goes down, the backup vLAG take over, by changing the vLAG port state from discarding to forwarding. The failover occurs within milliseconds. The port-channel redundancy group controls the port state of the member vLAGs to avoid looping errors.

FIGURE 20 Port-channel redundancy group configuration



There is no restriction on the membership of the active or backup vLAG in the fabric. Either the active or the backup vLAG can coexist in a single RBridge, or they can exist in different RBridges.

Consider the following guidelines and restrictions when configuring a port-channel redundancy group:

- The maximum number of supported port-channel redundancy groups is 255.
- Port-channel redundancy groups are supported only on port-channel interfaces.
- If you designate the active vLAG when configuring the port-channel redundancy group, it will always resume the active vLAG role after a failure, and the other vLAG will return to the backup role.
- If you do not designate one of the vLAGs as "active" then the system assigns the vLAG that comes first in the protected group as the active vLAG, and the second becomes the backup vLAG. If the active vLAG fails and the backup vLAG takes over, the backup vLAG will retain the active role when the original vLAG resets.
- Extreme trunks must not be a member of a protected group.
- Membership in a port-channel redundancy group cannot be altered when the group is in the active state. To modify the protected group or to change the "active" role, first deactivate the protected group.
- Port-channel redundancy groups do not support the STP family of protocols.

Configuring port-channel redundancy groups

Two port-channels are placed into a protected group. In this protected group, one port-channel acts as the active link, and the other port-channel acts as the standby link.

This task assumes that the port channels have already been created using the instructions in [Configuring vLAGs](#) on page 125.

To configure a port-channel redundancy group, perform the following task in global configuration mode.

1. Open the port-channel redundancy-group configuration mode with the **port-channel-redundancy-group** command.

```
switch(config)#port-channel-redundancy-group 27
switch(config-port-channel-redundancy-group-27)#
```

2. Add the active port-channel to the group with the **port-channel** command.

```
switch(config-port-channel-redundancy-group-27)# port-channel 3 active
```

3. Add the standby port-channel to the group with the **port-channel** command.

```
switch(config-port-channel-redundancy-group-27)# port-channel 5
```

4. Activate the port-channel redundancy group with the **activate** command.

```
switch(config-port-channel-redundancy-group-27)# activate
```

5. Confirm the configuration with the **show port-channel-redundancy-group** command.

```
switch#show port-channel-redundancy-group 27
Group ID                : 27
Member Ports           : Port-channel 3, Port-channel 5
Configured Active Port-channel: Port-channel 3
Current Active Port-channel  : Port-channel 3
Backup Port-channel     : Port-channel 5
```

Configuring an ESI on a port-channel for BGP routing

Ethernet Segment Identifiers (ESIs) are used to identify the links connecting multiple ToR devices to a server in a multihomed BGP EVPN environment. Use this task to configure ESI values manually on a port-channel interface if LACP autodiscovery is not deployed.

For details, see "Ethernet Segment Identifiers for BGP routing" earlier in this chapter.

1. From global configuration mode, specify a port-channel interface and enter port-channel configuration mode.

```
device# config terminal
device(config)# interface port-channel 1
device(config-Port-channel-1)#
```

2. Enter the **esi** command and specify an appropriate hexadecimal ESI value.

```
device(config-Port-channel-1)# esi 00:11:22:33:44:55:66:77:88:99
```

3. Repeat the above as appropriate on all port-channel interfaces to be configured manually.

4. Do any of the following to manage the configuration.

- a) Use the
- esi auto lacp**
- command to enable automatic ESI assignment.

```
device(config-Port-channel-1)# esi auto lacp
```

- b) Use the
- no esi<value>**
- command to remove the ESI value from the interface and allow a new number to be assigned.

```
device(config-Port-channel-1)# no esi 00:11:22:33:44:55:66:77:88:99
```

- c) Use the
- no esi auto lacp**
- command to reenble the assignment of the most current ESI to the interface.

```
device(config-Port-channel-1)# no esi auto lacp
```

Configuring load balancing on a remote RBridge

This feature allows you to configure the load-balancing feature on a remote RBridge, which is not a member of the vLAG (also known as a non-local RBridge), to forward traffic to a vLAG. To distribute the traffic among the possible paths towards the VLAG, you can configure the vLAG load-balancing flavor on RB2. Available flavors are listed below.

TABLE 12 Load balancing flavors

Flavor	Definition
dst-mac-vid	Destination MAC address and VID-based load balancing.
src-mac-vid	Source MAC address and VID-based load balancing.
src-dst-mac-vid	Source and Destination MAC address and VID-based load balancing.
src-dst-ip	Source and Destination IP address-based load balancing.
src-dst-ip-mac-vid	Source and Destination IP and MAC address and VID-based load balancing.
src-dst-ip-port	Source and Destination IP and TCP/UDP port-based load balancing.
src-dst-ip-mac-vid-port	Source and Destination IP, MAC address, VID and TCP/UDP port-based load balancing.

Additionally, an RBridge can be set to a different flavor for different vLAGs present in the cluster. This feature is available for each RBridge and each VLAG, so different load-balancing flavors can be set for traffic directed towards different VLAGs. The **show running-config rbridge-id rbridgeID** command displays the configuration information.

NOTE

When configuring load balancing on an Extreme VDX 6740, it should be configured consistently for all port-channels on the switch. These switches support one load-balancing scheme at a time, and apply the last loaded load-balancing scheme to all port-channels on the switch. This is not required for the Extreme VDX 8770 platform, as it supports multiple port-channel load-balancing schemes.

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

```
switch(config)# rbridge-id 2
switch(config-rbridge-id-2)# fabric port-channel 20 load-balance dst-mac-vid
switch(config-rbridge-id-2)# end
switch# show running-config rbridge-id 2
rbridge-id 2
 interface-nodespecific ns-vlan 10
 interface-nodespecific ns-ethernet 100
 fabric vlag 10 load-balance src-dst-mac-vid
 fabric vlag 20 load-balance dst-mac-vid
 no protocol vrrp
switch# show fabric port-channel load-balance 10
Fabric Vlag Load-Balance Information
-----
```

```
Rbridge-Id      : 2
Vlag           : 10
Load-Balance Flavor : Source and Destination MAC address and VID based load balancing
```

```
switch# show fabric port-channel all
```

```
Fabric Vlag Load-Balance Information
```

```
-----
Rbridge-Id      : 2
Vlag           : 10
```

Configuring and managing Link Aggregation

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

Understanding the default LACP configuration

The table below lists the default LACP configuration. Consider this when making changes to the defaults.

TABLE 13 Default LACP configuration

Parameter	Default setting
System priority	32768
Port priority	32768
Timeout	Long (standard LAG) or short (Extreme LAG)

Enabling LACP on an DCB interface

The switch must be in privileged EXEC mode.

To add additional interfaces to an existing LAG, repeat this procedure using the same LAG group number for the new interfaces.

To enable LACP on a DCB interface, perform the following steps.

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

```
switch(config)# interface tengigabitethernet 5/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.
4. Enter the **channel-group** command to configure the LACP for the DCB interface.

```
switch(conf-if-te-5/0/1)# channel-group 4 mode active type standard
```

Configuring neighbor discovery for LACP on an interface

You may need to disable neighbor discovery for Extreme devices on a per-interface basis so that the Extreme VDX does not to bring up its ports in an uncontrolled fashion until the fabric completely forms. This option is needed when an unconditional EtherChannel is configured between the VCS fabric and an end node, usually ESX or Hypervisors, which does not support LACP. If an Extreme VDX brings up its ports unexpectedly, the data traffic may be compromised.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface subtype configuration mode for the port.

```
device(config)# interface tengigabitethernet 1/0/18
```

3. Use the **fabric neighbor-discovery disable** command to disable neighbor discovery.

```
device(config-if-te-1/0/18)# fabric neighbor-discovery disable
```

4. Once the fabric has come online, use the **no fabric neighbor-discovery disable** command to reenable neighbor discovery on the interface.

```
device(config-if-te-1/0/18)# no fabric neighbor-discovery disable
```

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 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 a Data Center Bridging (DCB) interface, perform the following steps:

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

```
device(config)# interface tengigabitethernet 5/0/1
```

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

```
device(conf-if-te-5/0/1)# lacp timeout short
```

Configuring LACP for PXE

During the PXEBOOT stage, the PXE client remote MAC is learned against a port-channel interface in the device. Therefore, the traffic towards the Preboot eXecution Environment (PXE) client's remote MAC can go on all the links of the port-channel interface. But the PXE client picks only one link as active and drops the traffic that is received on all other links of the port-channel.

To handle this scenario, during PXEBOOT stage, you must configure **lACP default-up** on all the member ports of the port-channel and keep the member ports admin up. The **lACP default-up** command also enables the port-channel to be kept online, even when no LACP PDUs are received on a member link. After this configuration is complete and the port-channel comes online, the PXE client picks a link in that port-channel and sends traffic on that link. This generates the Layer 2 MAC learn event on that link of the port-channel. The device recognizes that it received Layer 2 MAC learn event on the "defaulted" port-channel. On getting this event, only that member link of the port-channel are kept online and other member ports are kept in "admin shut state" with exception. The traffic towards the PXE client is sent and received on only one member link of the port-channel. Hence the traffic drop on other members does not occur. Once the PXE client boots with its new downloaded image, the PXE client is able to exchange LACP PDUs to the device. Once this stage is attained, you must remove the LACP default-up config on all the member links of the port-channel. The device once receives a LACP PDU from the PXE client, LACP PDUs exchange happens and the VLAG comes up with all the configured links.

Use the **no lACP default-up** command to disable vLAG for PXE and fallback to the default LACP functionality.

1. Enter global configuration mode.

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

2. Configure the port-channel interface designated for PXE.

```
device(configure)# interface tengigabitethernet 1/0/9
```

3. Activate the channel group.

```
device(conf-if-te-1/0/9)# channel-group 10 mode active
```

4. Activate the LACP PXE.

```
device(conf-if-te-1/0/9)# lACP default-up
```

5. Repeat this task for all member interfaces of the port-channel.

Clearing LACP counter statistics on a LAG

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

To clear LACP counter statistics on a LAG, use the following command:

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.

To clear LACP counter statistics on all LAG groups, use the following command:

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

```
device# clear lacp counter
```

Displaying LACP information

You can use the **show** command in privileged EXEC mode to display Link Aggregation Control Protocol (LACP) information.

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

```
switch# show lacp sys-id
% System 8000,00-05-1e-76-1a-a6
```

- Enter the **show lacp counter** command to display the LACP system ID and priority.

```
switch# show lacp counter
Traffic Statistics
Port          LACPDU          Marker          Pckt err
           Sent      Recv          Sent      Recv          Sent      Recv
12            123        0              2         0             0         0
```

Refer to the *Extreme Network OS Command Reference* for more information.

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 the speed parameter is configured to 1000 if the port-channel is using the gigabit interface.
- 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.
- You can verify the system ID of the switches in the Extreme VCS Fabric cluster with the **show lacp sys-id** command.
- Make sure that LACPDU's are being received and transmitted on both ends of the link and that there are no error PDU's. 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."

If an Extreme-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 Extreme for 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. This can be verified 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 there are no error PDUs. This can be verified by entering the **show lacp counters number** command and looking at the rx and 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.
- Make sure that the fiber length of the link has a deskew value of 7 microseconds. If it does not, the link will not be able to join the LAG and the following RASlog message is generated: `Deskew calculation failed for link <link-name>`.

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

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

If an Extreme-based 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 Extreme 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.

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.

AMPP

- [AMPP overview.....](#)139
- [Configuring AMPP profiles.....](#)143

AMPP overview

Server virtualization infrastructure associates a server-side Virtual Ethernet Bridge (VEB) port-profile with each Ethernet MAC address used by a virtual machine (VM) to access the network through a VEB port. The Extreme Auto Migrating Port Profile (AMPP) feature provides advanced controls for maintaining and migrating these port-profile associations when a VM migrates across physical servers.

If the VM is migrated from one physical server to another, the VEB's port-profile migrates with it, providing automated port-profile migration of the server's VEB ports that are associated with the VM.

For environments where the server's virtualization infrastructure provides sufficient controls, automated port-profile migration approaches are fine. An example of such an environment is a high performance cluster that uses a Layer 2 network that is isolated from external networks through firewalls and security appliances.

However, there is a gap between the access and Quality of Service (QoS) controls supported in external Layer 2 switches and the server virtualization infrastructure. External Layer 2 switches have more advanced controls compared to server VEB implementations.

Some environments prefer the more advanced controls provided by external network switches. An example of such an environment is a multi-tier data center that has several types of applications, each with differing advanced network controls, running over the same Layer 2 network. In this type of environment, the network administrator often prefers the use of advanced access controls available in external switches.

Layer 2 networks do not provide a mechanism for automatically migrating switch access and traffic controls associated with an end-point device when that device migrates from one switch to another. The migration may be physical, such as an operating system image (such as an application, middleware, operating system, and associated state) that is running BareMetal OS on one system and is migrated to another system. The migration may be also be virtual, such as an operating system image that is running over Hypervisor VMware on one system and is migrated to run over Hypervisor VMware on another system.

AMPP over vLAG

Virtual Link Aggregation Group (vLAG) is the name for Extreme proprietary LAG in which the links to the Extreme VCS Fabric can be connected to one or more physical switches or servers. For redundancy and greater bandwidth, vLAG is a vital component of Extreme VCS Fabric technology. AMPP is supported on vLAG and standard LAG in a manner similar to physical port.

FCoE capability on all port-profiled interfaces can be activated using the `fcoe-port` configuration in the default port-profile (refer to [Configuring FCoE profiles](#) on page 145). This configuration enforces FCoE capability only on physical interfaces, not on the port-channel LAG. Member links of the LAG must be explicitly configured for FCoE capability.

For complete information on vLAG, refer to [Link Aggregation Control Protocol](#) on page 118.

The *italic* text in the following example highlights the vLAG information in the port profile:

```
switch# show port-profile status

Port-Profile      PPID      Activated  Associated MAC  Interface
auto-dvPortGroup  1         Yes       None           None
auto-dvPortGroup2 2         Yes       None           None
auto-dvPortGroup3 3         Yes       None           None
auto-dvPortGroup_4_0 4         Yes       0050.567e.98b0 None
```

auto-dvPortGroup_vlag	5	Yes	0050.5678.eaed	None
auto-for_iscsi	6	Yes	0050.5673.85f9	None
			0050.5673.fc6d	None
			0050.5674.f772	None
			0050.5675.d6e0	Te 234/0/54
			0050.567a.4288	None
auto-VM_Network	9	Yes	000c.2915.4bdc	None
			0050.56a0.000d	None
			0050.56a0.000e	None
			0050.56a0.000f	None
			0050.56a0.0010	Po 53
			0050.56a0.0011	Po 53
			0050.56a0.0012	Po 53
			0050.56a0.0013	None
			0050.56a0.0025	None
			0050.56a0.0026	None
			0050.56a0.0027	None
			0050.56a0.0028	None
			0050.56a0.0029	Po 53
			0050.56a0.002a	Po 53
			0050.56a0.002b	Po 53
			0050.56a0.002c	None
			0050.56a0.002d	None
			0050.56a0.002e	None
			0050.56a0.002f	None
			0050.56b3.0001	Po 53
			0050.56b3.0002	Po 53
			0050.56b3.0004	Po 53
			0050.56b3.0005	None
auto-VM_kernel	10	Yes	0050.5671.4d06	None
			0050.5672.862f	Po 53
			0050.5678.37ea	None
			0050.567a.ddc3	None
auto-VM_NW_1G	11	Yes	0050.56b3.0000	None
			0050.56b3.0003	Po 82
			0050.56b3.0007	None
			0050.56b3.0008	Po 82
			0050.56b3.0009	Po 82
auto-VMkernel	12	Yes	0050.567a.fdcf	Po 82
			0050.567c.c2e3	None
auto-VMkernel_VS	13	Yes	0050.567d.16b9	None
			0050.567e.e25b	None
auto-Management+Network	14	Yes	5cf3.fc4d.ca88	None
auto-Virtual+Machine+Network	15	Yes	000c.2941.27e2	None
			000c.2980.335d	None

```

switch# show port-profile int all
Interface      Port-Profile
Gi 234/0/1     None
Gi 234/0/13    None
Gi 234/0/25    None
Gi 234/0/26    None
Te 234/0/54    auto-for_iscsi
Po 82          auto-VM_NW_1G
               auto-VMkernel
Po 53          auto-VM_Network
               auto-VM_kernel

```

AMPP and Switched Port Analyzer

Switched Port Analyzer (SPAN), or Port Mirroring, selects network traffic for analysis by a network analyzer. If you are interested in listening or snooping on traffic that passes through a particular port, Port Mirroring is necessary to artificially copy the packets to a port connected to the analyzer. However, SPAN and the Auto Migrating Port Profile (AMPP) port-profile-port configuration are mutually exclusive. All of the Layer 2 configuration for SPAN is mutually exclusive with regard to the **port-profile-port** command for AMPP.

AMPP scalability

The following table describes the Auto Migrating Port Profile (AMPP) scalability values supported by Network OS 5.0.0 and later.

TABLE 14 AMPP scalability values

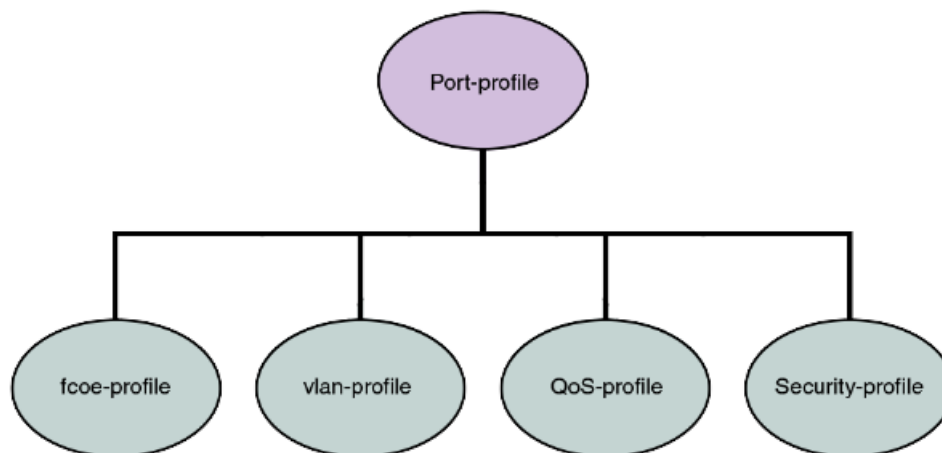
Metric	Value
Number of profiles	1000
Number of VLANs in port-profiles	2000
QoS profile	1 cee-map 1 mutation-map
Number of ACLs in security profiles	1 ingress MAC ACL 1 egress MAC ACL 1 ingress IPv4 ACL 1 egress IPv4 ACL 1 ingress IPv6 ACL 1 egress IPv6 ACL

For the number of MAC associations that are supported, refer to the Release Notes. The practical number of MAC associations and VLANs that are supported will vary depending on the ACL configuration. In addition, AMPP is subject to the maximum number of vLAGs and LAGs supported on the switch, which is 1000 in this case.

AMPP port-profiles

As shown in the following figure, the default port-profile contains the entire configuration needed for a VM to get access to the LAN and SAN.

FIGURE 21 Port-profile contents



In addition, all the combinations can be mixed up with some security rules grouped under a security-profile.

NOTE

A port-profile does not contain some of the interface level configurations, such as LLDP, SPAN, LAG, and so on.

A port-profile operates as a self-contained configuration container. In other words, if a port-profile is applied on a completely new switch without any configuration, it is capable of configuring the interface's local configuration and starting to carry traffic. Any changes to the policies are immediately applied to the data plane.

Security profiles are applied to the ACLs based on the profile or PolicyID. Therefore, multiple security profiles can be applied to the same profiled port.

NOTE

The fcoe-profile is available for both default and nondefault profiles. User-defined port-profiles do not have access to the fcoe-profile. However, editing of the port-profile is not allowed once the port-profile is activated. Activation of the port-profile is mandatory when it is applied to a port. Refer to [Configuring FCoE profiles](#) on page 145 for additional details.

Life of a port-profile

A port-profile during creation will go through multiple states. The states of a port-profile are as follows:

- **Created** —This state specifies that a port-profile is created or modified, but may not be complete.
- **Activated** —This state specifies that a port-profile is activated and is available for MAC->port-profile association. If the port-profile created is not complete then the activation fails; you must resolve any conflicts or dependencies and reactivate the port-profile.
- **Associated** —This state specifies that one or more MAC addresses have been associated to this port-profile within the fabric.
- **Applied** —This state indicates that the port-profile is applied on the profiled port where the associated MAC address appeared. In the absence of any signaling protocol, the system snoops the packet to detect if the associated MAC address has appeared on the profiled port. Configuration of two different port-profiles can co-exist on a profiled port, but if there is a conflict then the application of the later port-profile fails.

The following table describes the AMPP events and the applicable failure behaviors.

TABLE 15 AMPP behavior and failure descriptions

AMPP event	Applicable behavior and failures
Create port-profile	<ul style="list-style-type: none"> • If the port-profile does not exist, then it is created. If it exists, then it is available for modification (if it is not yet activated).
Activate port-profile	<ul style="list-style-type: none"> • If the port-profile configuration is not complete, activation fails. Unless the port-profile is activated, it is not applied on any port-profile-port. • If all the dependency validations succeed, the port-profile is in the ACTIVE state and is ready for association. • A vlan-profile is mandatory for all port-profiles.
De-activate port-profile	<ul style="list-style-type: none"> • This event removes the applied port-profile configuration from all the profiled-ports. • De-activation is allowed even if there are MAC addresses associated with the port-profile.
Modify port-profile	<ul style="list-style-type: none"> • Port-profile can be edited only in the pre-activation stage. • The port-profile is set to the INACTIVE state if any conflicting attributes are configured, or some dependent configuration is not completed. • Port-profile state is set as INACTIVE and any attempt to associate the port-profile to a MAC address may not be allowed.
Associate MAC addresses to a port-profile	<ul style="list-style-type: none"> • If the MAC is already associated with a port-profile, the port-profile to MAC association fails. • Otherwise, if the port-profile to MAC association succeeds, when the same MAC is learned on any of the ports, the port-profile which has the MAC association is applied to the port.

TABLE 15 AMPP behavior and failure descriptions (continued)

AMPP event	Applicable behavior and failures
De-associate MAC addresses from a port-profile	<ul style="list-style-type: none"> If mapping exists, all the policies configured for a specific MAC address are removed from that port or switch.
Deleting a port-profile	<ul style="list-style-type: none"> An IN USE error is generated if the port-profile is in an activated state. AMPP forces you to deactivate the profile before deleting. If the port-profile is in an inactive state, then deletion of profile removes all the MAC associations as well.
Modifying port-profile content when in an associated state	<ul style="list-style-type: none"> An IN USE error is generated if the port-profile is already activated.
Moving the VM MAC and notifying the fabric	<ul style="list-style-type: none"> All policies associated to the port-profile ID are mapped on the MAC address and applied to the new port in the fabric.
Unused port-profile	<ul style="list-style-type: none"> You must manually remove the port-profile to MAC associations.

Configuring AMPP profiles

The following sections cover configuring, deleting, and monitoring various AMPP-related profiles.

Configuring a new port-profile

To support VM MAC address learning, the default port-profile is employed. The default profile is different from the other user-defined AMPP profiles:

- The **allow non-profiled-macs** command only functions with the default port-profile. Refer to the *Extreme Network OS Command Reference*.
- The **restrict-flooding** command only functions with the default port-profile. Refer to the *Extreme Network OS Command Reference*.
- The port-profile ID (ppid) of the profile cannot be changed.
- The VLAN sub-profile cannot be modified.
- The QoS sub-profile and security-profile cannot be added.
- The default port-profile cannot be activated.

Extreme recommends that you create a new port-profile to accommodate your requirements. To configure a new port-profile, perform the following steps in privileged EXEC mode.

1. Configure the physical interface, LAG, or vLAG as a port-profile port.

```
switch(if-te-2/0/1)# port-profile-port
```

2. Create and configure a new port-profile name.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# port-profile vml-port-profile
switch(config-port-profile-vml-port-profile)# vlan-profile
switch(config-pp-vlan)# switchport
switch(config-pp-vlan)# switchport mode trunk
switch(config-pp-vlan)# switchport trunk native-vlan 300
switch(config-pp-vlan)# switchport trunk allowed vlan add 300
```

- Exit VLAN profile configuration mode.

```
switch(config-pp-vlan)# exit
```

- Activate the profile.

```
switch(config)# port-profile vml-port-profile activate
```

- Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vml-port-profile static 0050.56bf.0001
switch(config)# port-profile vml-port-profile static 0050.56bf.0002
switch(config)# port-profile vml-port-profile static 0050.56bf.0003
switch(config)# port-profile vml-port-profile static 0050.56bf.0004
switch(config)# port-profile vml-port-profile static 0050.56bf.0005
```

Configuring VLAN profiles

The VLAN profile defines the VLAN membership of the overall port-profile, which includes both the tagged and untagged VLANs.

NOTE

Private VLAN port mode commands are not available for AMPP VLAN profiles.

To configure the VLAN profile, perform the following steps in global configuration mode.

- AMPP profiles cannot be modified while active. De-activate the port-profile before modifying the VLAN profile.

```
switch(config)# no port-profile vml-port-profile activate
```

- Enter VLAN profile configuration mode.

```
switch(config)# port-profile vml-port-profile
switch(config-port-profile-vml-port-profile)# vlan-profile
```

- Use the **switchport** command to change the mode to Layer 2 and set the switching characteristics to the defaults.

```
switch(config-pp-vlan)# switchport
```

- Access the VLAN profile mode for the correct VLAN.

```
switch(config-pp-vlan)# switchport access vlan 200
```

- Enter trunk configuration mode.

```
switch(config-pp-vlan)# switchport mode trunk
```

- Configure the trunk mode for the allowed VLAN IDs.

```
switch(config-pp-vlan)# switchport trunk allowed vlan add 10, 20, 30-40
```

- Configure the trunk mode to be a native VLAN.

```
switch(config-pp-vlan)# switchport trunk native-vlan 300
```

- Exit VLAN profile configuration mode.

```
switch(config-pp-vlan)# exit
```


9. Activate the profile.

```
switch(config)# port-profile vml-port-profile activate
```

10. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vml-port-profile static 0050.56bf.0001
switch(config)# port-profile vml-port-profile static 0050.56bf.0002
switch(config)# port-profile vml-port-profile static 0050.56bf.0003
switch(config)# port-profile vml-port-profile static 0050.56bf.0004
switch(config)# port-profile vml-port-profile static 0050.56bf.0005
```

Configuring FCoE profiles

Both default and nondefault port profiles are supported. Refer to [FCoE](#) on page 13 for details.

NOTE

Before a port profile can be modified, no interfaces can have a port-profile-port configuration.

In the absence of the FCoE profile in the default AMPP profile, you can configure FCoE on a per-interface basis, based on the profiled ports.

To configure a default FCoE profile globally, perform the following steps in global configuration mode.

1. Enter port-profile configuration mode.

```
switch(config)# port-profile default
```

2. Enter FCoE-profile configuration mode.

```
switch(config-port-profile-default)# fcoe-profile
```

3. Activate the FCoE port profile.

An FCoE map cannot be applied on interfaces that already have a CEE map applied to it.

```
switch(config-fcoe-profile)# fcoeport default
```

Configuring QoS profiles

QoS profiles define the following values:

- Incoming 802.1p priority is set to internal queue priority. If the port is in QoS untrusted mode, all incoming priorities will be mapped to default best effort priority.
- Incoming priority is set to outgoing priority.
- Mapping of incoming priorities is set to strict or WRR traffic classes.
- Enabling of flow control on a strict or a WRR traffic class.

The QoS profile has two flavors: CEE QoS and Ethernet QoS. The QoS profile may contain either CEE QoS or Ethernet QoS. Server side ports typically are carrying converged traffic.

To configure the QoS profile, perform the following steps in global configuration mode.

1. AMPP profiles cannot be modified while active. Deactivate the port-profile before modifying the VLAN profile.

```
switch(config)# no port-profile vml-port-profile activate
```

2. Enter QoS profile mode.

```
switch(config)# port-profile vml-port-profile
switch(config-port-profile-vml-port-profile)# qos-profile
switch(config-qos-profile)#
```

3. Apply the CEE map.

```
switch(config-qos-profile)# cee default
```

4. Apply a map to the profile. You can do either of the following:

- Apply the existing CoS-to-CoS mutation map.

```
switch(config-qos-profile)# qos cos-mutation vml-cos2cos-map
```

- Apply the existing CoS-to-Traffic-Class map.

```
switch(config-qos-profile)# qos cos-traffic-class vml-cos2traffic-map
```

5. Enable pause generation for each CoS.

```
switch(config-qos-profile)# qos flowcontrol tx on rx on
```

6. Exit QoS profile mode.

```
switch(config-qos-profile)# exit
```

7. Activate the profile.

```
switch(config)# port-profile vml-port-profile activate
```

8. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vml-port-profile static 0050.56bf.0001
switch(config)# port-profile vml-port-profile static 0050.56bf.0002
switch(config)# port-profile vml-port-profile static 0050.56bf.0003
switch(config)# port-profile vml-port-profile static 0050.56bf.0004
switch(config)# port-profile vml-port-profile static 0050.56bf.0005
```

Configuring security profiles

A security profile defines all the security rules needed for the server port. A typical security profile contains attributes for MAC-based and IP-based standard and extended ACLs. Security profiles are applied to the ACLs based on the profile or PolicyID. Therefore, multiple security profiles can be applied to the same profiled port.

To configure the security profile, perform the following steps in global configuration mode.

1. AMPP profiles cannot be modified while active. Deactivate the port-profile before modifying the security profile.

```
switch(config)# no port-profile vml-port-profile activate
```

2. Enter security profile configuration mode.

```
switch(config)# port-profile vml-port-profile
switch(config-pp)# security-profile
switch(config-pp-security)#
```

3. Modify the ACL security attributes. Refer to the *Network OS Security Guide* for details on modifying ACLs.

4. Apply the ACL to the security profile.

```
switch(config-pp-security)# mac access-group vml-acl in
```

5. Exit security profile configuration mode.

```
switch(config-pp-security)# exit
```

6. Activate the profile.

```
switch(config)# port-profile vml-port-profile activate
```

7. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vml-port-profile static 0050.56bf.0001
switch(config)# port-profile vml-port-profile static 0050.56bf.0002
switch(config)# port-profile vml-port-profile static 0050.56bf.0003
switch(config)# port-profile vml-port-profile static 0050.56bf.0004
switch(config)# port-profile vml-port-profile static 0050.56bf.0005
```

Creating a port-profile-port

To create a port-profile-port, perform the following steps in global configuration mode.

1. Activate the interface configuration mode for the interface you wish to modify.

The following example activates the mode for the 10-gigabit Ethernet interface in slot 0/port 0.

```
switch(config)# interface tengigabitethernet 1/0/1
```

2. Configure port-profile-port on the physical interface.

```
switch(conf-int-te-1/0/1)# port-profile-port
```

Deleting a port-profile-port

To delete a port-profile-port, perform the following steps in global configuration mode.

1. Activate the interface configuration mode for the interface you wish to modify.

The following example activates the mode for the 10-gigabit Ethernet interface in slot 0/port 0.

```
switch(config)# interface tengigabitethernet 1/0/1
```

2. Unconfigure port-profile-port on the physical interface.

```
switch(conf-int-te-1/0/1)# no port-profile-port
switch(conf-int-te-1/0/1)# no shutdown
```

Deleting a port-profile

To delete a port-profile, perform the following steps in privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

2. Deactivate the port-profile.

```
switch(config)# no port-profile vml-port-profile activate
```

3. Use the **no** form of the **port-profile** command to delete the custom profile.

You cannot delete the default port-profile.

```
switch(config)# no port-profile vml-port-profile
```

Deleting a sub-profile

To delete a sub-profile, perform the following steps in privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

2. Deactivate the port-profile.

```
switch(config)# no port-profile vml-port-profile activate
```

3. Enter port-profile mode.

```
switch(conf-vml-port-profile)# port-profile vml-port-profile
```

4. Use the following to delete sub-profiles.

- To delete a VLAN sub-profile:

```
switch(conf-vml-port-profile)# no vlan-profile
```

- To delete a security sub-profile:

```
switch(conf-vml-port-profile)# no security-profile
```

- To delete a FCoE sub-profile under default profile:

```
switch(conf-pp-default)# no fcoe-profile
```

- To delete a QoS sub-profile:

```
switch(conf-vml-port-profile)# no qos-profile
```

Creating a new port-profile domain and adding port profiles

Use the **port-profile-domain** command to create an AMPP port-profile domain that contains all of the port profiles that can be applied to a profiled port in a Virtual Fabrics context.

1. In global configuration mode, create a port-profile domain.

```
device(config)# port-profile-domain my_PP_domain
```

2. In port-profile-domain configuration mode, use the **port-profile** command to add profiles to that domain.

```
device(config-port-profile-domain-my_PP_domain)# port-profile my_PP_1
device(config-port-profile-domain-my_PP_domain)# port-profile my_PP_2
```

Monitoring AMPP profiles

To monitor the AMPP profiles, perform the following steps in privileged EXEC mode.

1. Use the **show** command to display the current MAC details.

```
switch# show mac-address-table port-profile
Legend: Untagged(U), Tagged (T), Not Forwardable(NF) and Conflict(C)
VlanId  Mac-address      Type      State      Port-Profile  Ports
1       0050.5679.5351   Dynamic   Active     Profiled(U)   Te 111/0/10
1       0050.567b.7030   Dynamic   Active     Profiled(U)   Te 111/0/12
1       005a.8402.0000   Dynamic   Active     Profiled(T)   Te 111/0/24
1       005a.8402.0001   Dynamic   Active     Profiled(NF)  Te 111/0/24
1       005a.8402.0002   Dynamic   Active     Not Profiled  Te 111/0/24
1       005a.8402.0003   Dynamic   Active     Not Profiled  Te 111/0/24
1       005a.8402.0004   Dynamic   Active     Not Profiled  Te 111/0/24
1       005a.8402.0005   Dynamic   Active     Profiled(NF)  Te 111/0/24
1       005a.8402.0006   Dynamic   Active     Not Profiled  Te 111/0/24
1       005a.8402.0007   Dynamic   Active     Profiled(T)   Te 111/0/24
1       005b.8402.0001   Dynamic   Active     Profiled(T)   Te 111/0/24
1       005c.8402.0001   Dynamic   Active     Profiled(T)   Te 111/0/24
100     005a.8402.0000   Dynamic   Active     Profiled      Te 111/0/24
100     005a.8402.0001   Dynamic   Active     Profiled(NF)  Te 111/0/24
100     005a.8402.0003   Dynamic   Active     Not Profiled  Te 111/0/24
100     005a.8402.0005   Dynamic   Active     Profiled(NF)  Te 111/0/24
100     005a.8402.0007   Dynamic   Active     Profiled      Te 111/0/24
Total MAC addresses : 17
```

2. Use the **show running-config** command to display all the available port-profile configurations.

```
switch# show running-config port-profile
port-profile default
vlan-profile
switchport
switchport mode trunk
switchport trunk allowed vlan all
switchport trunk native-vlan 1
!
fcoe-profile
fcoeport default
!
!
port-profile pp1
vlan-profile
switchport
switchport mode access
switchport access vlan 1
!
qos-profile
!
!
port-profile pp1 activate
port-profile pp1 static 1000.0000.0001
```

- Use the **show port-profile** command to display the current port-profile configuration.

```
switch# show port-profile
port-profile default
ppid 0
vlan-profile
switchport
switchport mode trunk
switchport trunk native-vlan 1
port-profile UpgradedVlanProfile
ppid 1
vlan-profile
switchport
switchport mode trunk
switchport trunk allowed vlan all
```

- Use the **show port-profile status** command to display the current status of all AMPP profiles.

```
switch# show port-profile status applied
Port-Profile      PPID  Activated  Associated MAC  Interface
auto-for_iscsi    6     Yes        0050.5675.d6e0 Te 9/0/54
auto-VM_Network   9     Yes        0050.56b3.0001 Te 9/0/53
                  0050.56b3.0002 Te 9/0/53
                  0050.56b3.0004 Te 9/0/53
                  0050.56b3.0014 Te 9/0/53

switch# show port-profile status activated
Port-Profile      PPID  Activated  Associated MAC  Interface
auto-dvPortGroup  1     Yes        None            None
auto-dvPortGroup2 2     Yes        None            None
auto-dvPortGroup3 3     Yes        None            None
auto-dvPortGroup_4_0 4     Yes        0050.567e.98b0 None
auto-dvPortGroup_vlag 5     Yes        0050.5678.eaed None
auto-for_iscsi     6     Yes        0050.5673.85f9 None

switch# show port-profile status associated
Port-Profile      PPID  Activated  Associated MAC  Interface
auto-dvPortGroup_4_0 4     Yes        0050.567e.98b0 None
auto-dvPortGroup_vlag 5     Yes        0050.5678.eaed None
auto-for_iscsi     6     Yes        0050.5673.85f9 None
```

- Use **show port-profile interface all** to display profile and applied interface information.

```
switch# show port-profile interface all
Port-profile      Interface
auto-VM_Network   Te 9/0/53
auto-for_iscsi    Te 9/0/54
```

Link-State Tracking (LST)

- Link-State Tracking (LST) overview..... 151
- General configuration guidelines for LST 153
- Configuring LST for independent RBridges..... 153
- Configuring LST for VCS fabrics..... 156
- Disabling LST..... 160
- LST show commands..... 161

Link-State Tracking (LST) overview

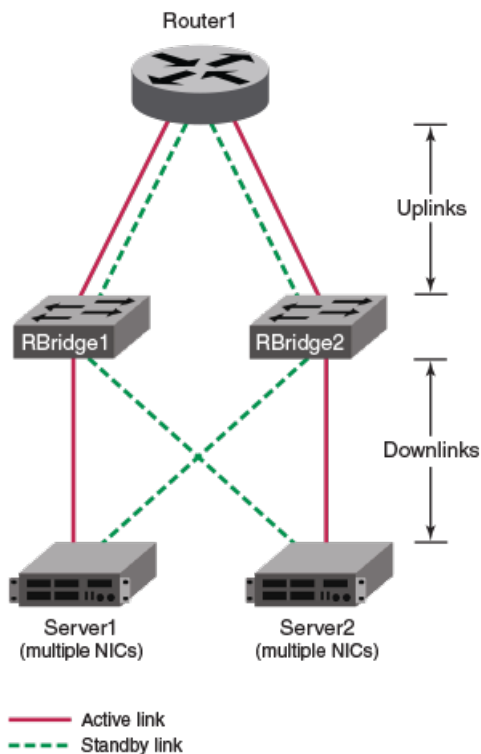
Link-state tracking (LST) is a relationship that you can configure and enable for redundant-link networking topology, to prevent traffic loss between upstream and downstream RBridge links.

Redundant-link topology

In a redundant-link topology, an alternate, standby network path is defined. If the primary, active path becomes unavailable, network traffic is rerouted to the standby path.

The following is a basic redundant-link topology:

FIGURE 22 A redundant-link topology



The primary path from the NICs of Server1 to Router1 is through RBridge1, and the alternate path is through RBridge2.

The primary path from the NICs of Server2 to Router1 is through RBridge2, and the alternate path is through RBridge1.

Redundant-link topology can include the following options:

- Links can be any of the following types:
 - Single physical port
 - Multiple physical ports
 - Port-channels
 - Breakout ethernet port
- VCS fabrics can be in place of or in addition to the individual RBridges. For more details, refer to [VCS redundant-link topology](#) on page 157.

If an upstream device stops functioning, there is a danger that the RBridge will continue to forward traffic upstream on the primary path, with ensuing loss of data.

LST operation

When link-state tracking (LST) is enabled, if an upstream link goes down, the downstream link automatically shuts down. At that point, server traffic is routed through the standby path, with negligible traffic loss. If the primary upstream link returns, LST restores the primary downstream link and reroutes the upstream traffic through the primary path.

The following upstream issues are among those that LST can handle:

- Failure of an upstream device
- Cable disconnection
- Other link failure

NOTE

LST cannot track an upstream link in a remote Rbridge.

Default response to uplink failure

As indicated in the following table, LST supports various combinations of uplinks and downlinks. By default, if one or more tracked uplinks fail, LST shuts all downlinks.

TABLE 16 Default response to uplink failure

Downstream links	Upstream links	Default response to uplink failure
1	1	If the uplink fails, LST shuts the downlink.
n	1	If the uplink fails, LST shuts all downlinks.
1	n	If one or more uplinks fail, LST shuts the downlink.
n	n	If one or more uplinks fail, LST shuts all downlinks.

NOTE

When considering the number of uplinks or downlinks, a port-channel is equivalent to a physical port.

Response to uplink failure with min-link specified

For multiple uplinks, the **track** command **min-link** option enables you to modify the default LST response to uplink failure.

By default, if one or more tracked uplinks fail, LST shuts all downlinks that track those uplinks. But if you specify **min-link**, such downlinks remain open if the number of functioning uplinks is equal to or greater than **min-link**.

As indicated in the following table, for multiple uplinks the default response is modified by the **min-link** value.

TABLE 17 Response to uplink failure with min-link specified

Downstream links	Upstream links	Response to uplink failure with min-link specified
1	1	If the uplink fails, LST shuts the downlink.
n	1	If the uplink fails, LST shuts all downlinks.
1	n	If the number of functioning uplinks < min-link, LST shuts the downlink.
n	n	If the number of functioning uplinks < min-link, LST shuts all downlinks.

General configuration guidelines for LST

The following are general configuration guidelines for link-state tracking (LST):

- LST is supported only for the following Layer 2 or Layer 3 interface types:
 - Physical interfaces (including breakout ports)
 - Port channels
 - > However, port channels are not supported for LST under FlexPort.
 - > For a port-channel downlink, if an upstream link goes down, LST shuts down the member physical ports of the port channel, but not the port channel.
- The following are examples of interface types for which LST is NOT supported:
 - 100 Mbps ports
 - Fibre-channel (FC)
 - Management (including loopback)
 - Switch virtual interface (SVI)
- For multiple uplinks and multiple downlinks, LST is supported also for configurations that include both Layer 2 and Layer 3 ports.
- You cannot configure a given port both as an LST uplink and downlink.
- In general, LST is configured on primary links, but not on standby links.
- You can implement LST on multiple hops in a network.
- You can implement LST on forward-referenced port channels (LAG ports with no member ports).
- LST operates on the operational level of uplinks rather than their STP forwarding state. So the following are required:
 - The redundant network must be loop-free.
 - If STP or RSTP is enabled, every uplink under LST must be in STP forwarding state.

NOTE

If needed, refer also to [LST configuration guidelines under VCS](#) on page 158.

Configuring LST for independent RBridges

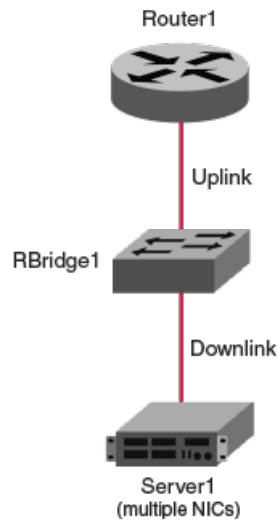
These topics explain how to configure link-state tracking (LST) for non-VCS topologies.

Configuring LST for single-link topologies

Use this procedure to configure LST on an RBridge with one uplink and one downlink.

In the following diagram of a single-link network topology, any of the links can be either physical port or port-channel.

FIGURE 23 Single-link topology



1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

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

3. Enter the **track interface** command to configure tracking for an uplink interface.

```
device(conf-if-te-1/0/1)# track interface ethernet 1/0/20
```

4. Enter the **track enable** command to enable tracking.

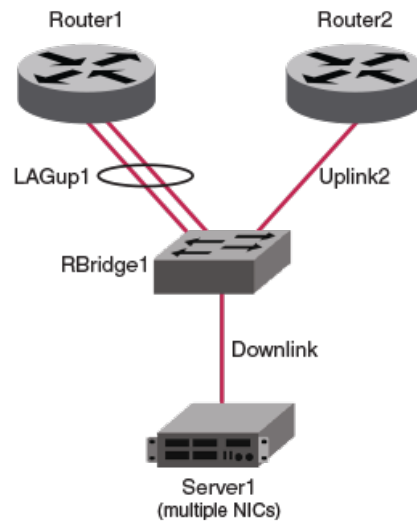
```
device(conf-if-te-1/0/1)# track enable
```

Configuring LST for multiple-uplink topologies

Use this procedure to configure LST on an RBridge with one downlink and two or more uplinks.

In the following diagram of a single-downlink, multiple-uplink topology, any of the links can be either physical port or port-channel.

FIGURE 24 Single-downlink, multiple uplink topology



1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 3/0/8
```

3. For each uplink interface that you want to track, enter the **track interface** command.

```
device(conf-if-te-3/0/8)# track interface ethernet 3/0/18
device(conf-if-te-3/0/8)# track interface ethernet 3/0/19
```

4. To modify the default behavior (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(conf-if-te-3/0/8)# track min-link 1
```

In this case, the downlink shuts down only if all of the uplinks are down. If only one of the two uplinks are down, the downlink stays open.

5. Enter the **track enable** command to enable tracking.

```
device(conf-if-te-3/0/8)# track enable
```

Configuring LST for multiple downlink/uplink topologies

Use this procedure to configure LST on an RBridge with multiple downlinks and multiple uplinks.

Perform this procedure on every downlink interface for which you are implementing LST. For example, on Switch3 in the below diagram, perform this procedure on the following interfaces:

- LAGdown1
- Downlink2

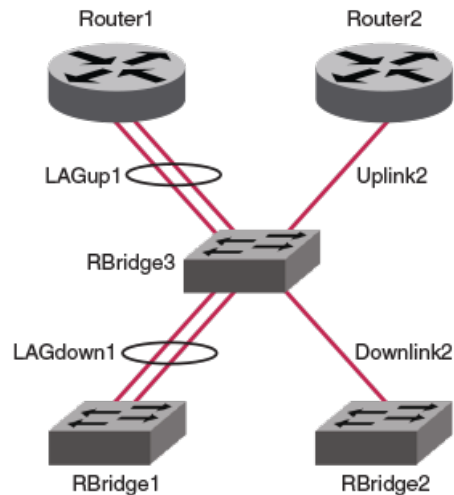
The code examples in this procedure are performed on LAGdown1, which tracks two uplink interfaces:

- LAGup1 (a port-channel)

- Uplink2 (a physical port)

In the following diagram of a multiple-downlink, multiple-uplink topology, any of the links can be either physical port or port-channel.

FIGURE 25 Multiple downlink/uplink topology



1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 3/0/8
```

3. For each uplink interface that you want to track, enter the **track interface** command.

```
device(conf-if-te-3/0/8)# track interface ethernet 3/0/18
device(conf-if-te-3/0/8)# track interface port-channel 1
```

4. To modify the default behavior (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(conf-if-te-3/0/8)# track min-link 1
```

In this case, the downlink shuts down only if all of the uplinks are down. If only one of the two uplinks are down, the downlink stays open.

5. Enter the **track enable** command to enable tracking.

```
device(conf-if-te-3/0/8)# track enable
```

Configuring LST for VCS fabrics

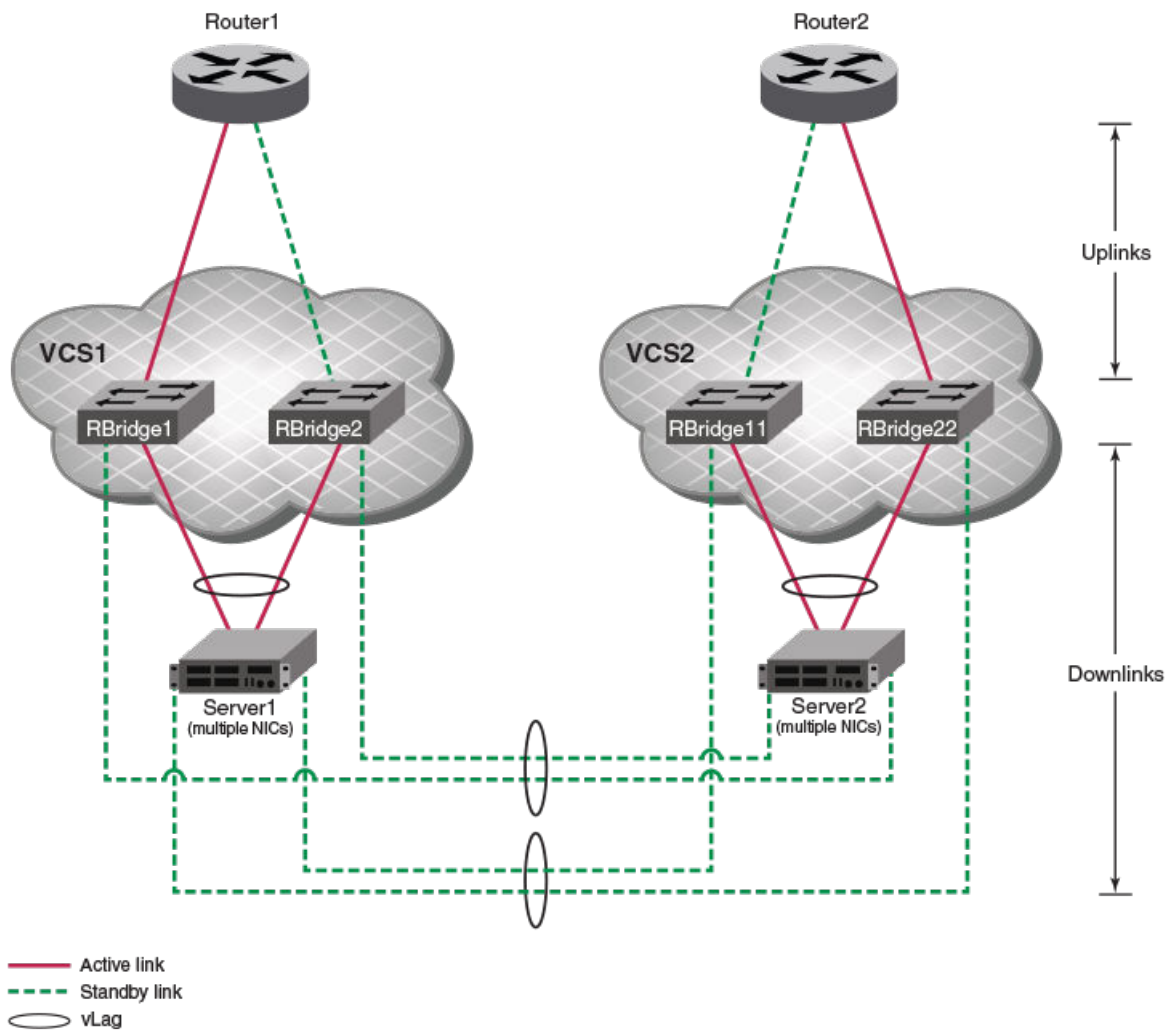
These topics explain how to configure link-state tracking (LST) for topologies that include one or more VCS fabrics.

VCS redundant-link topology

In a redundant-link topology, an alternate, standby network path is defined. If the primary, active path becomes unavailable, network traffic is rerouted to the standby path. This section deals with redundant-link topologies that include one or more VCS fabrics.

The following is a redundant-link topology that includes multiple VCS fabrics.

FIGURE 26 Redundant-link topology for VCS fabrics



The primary path from the NICs of Server1 to Router1 is through VCS1, and the alternate path is through VCS2.

The primary path from the NICs of Server2 to Router1 is through VCS2, and the alternate path is through VCS1.

LST configuration guidelines under VCS

The following are additional configuration guidelines for LST on R Bridges in VCS fabrics.

NOTE

Refer also to [General configuration guidelines for LST](#) on page 153.

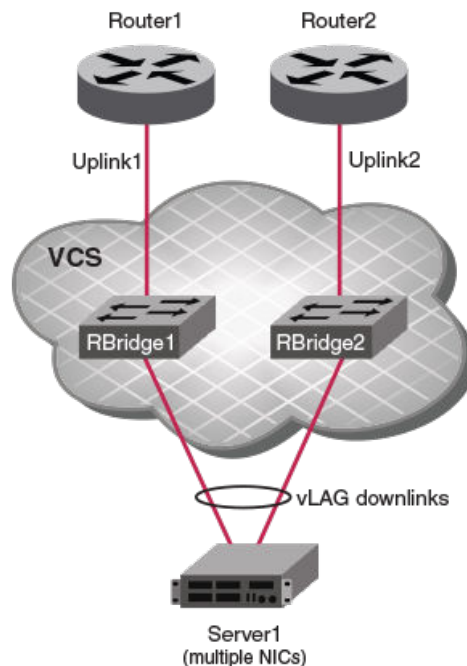
- In a VCS cluster, only local RBridge uplinks and downlinks are supported for LST; remote ports (even within the same VCS) are not supported.
- In a VCS cluster, only Ethernet FlexPorts are supported for LST. Fibre channel FlexPorts are not supported.

Configuring LST on a VCS cluster

Use this procedure to configure LST on a topology that includes a VCS but no independent R Bridges.

In this procedure, the downlink from the VCS is a vLAG and the uplink is a physical port. You can modify this procedure for related topologies.

FIGURE 27 VCS cluster for LST implementation



1. Log in to the VCS principal RBridge.
2. Enter **configure** to access global configuration mode.


```
device# configure
```
3. Enter the **interface** command to access the downlink interface.

```
device(config)# interface port-channel 1
```

- For each uplink interface that you want to track, enter the **track interface** command.

```
device(config-port-channel-1)# track interface ethernet 1/0/10
device(config-port-channel-1)# track interface ethernet 1/0/11
```

- To modify the default behavior under multiple uplinks on VCS R Bridges (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(config-port-channel-1)# track min-link 1
```

In this case, the downlink vLAG member port on RBridge1 shuts down only if Uplink1 goes down. Similarly, the downlink vLAG member port on RBridge2 shuts down only if Uplink2 goes down.

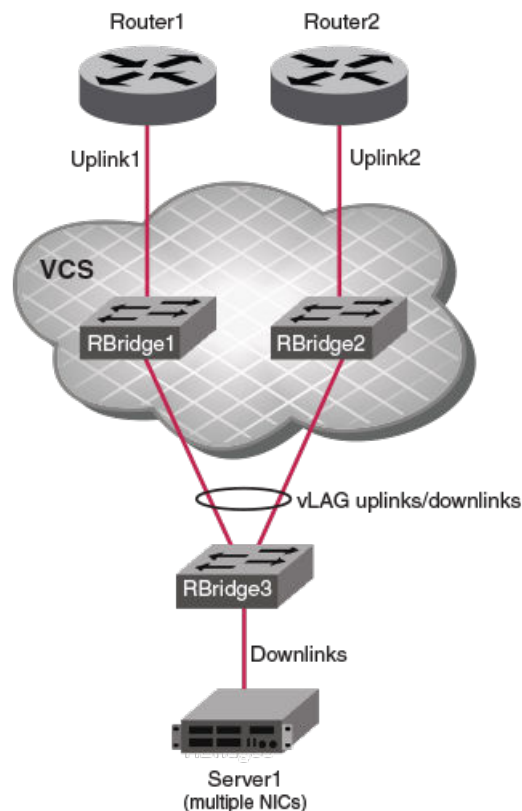
- Enter the **track enable** command to enable tracking.

```
device(config-port-channel-1)# track enable
```

Configuring LST on a VCS cluster and an independent RBridge

Use this procedure to configure LST on a topology that includes a VCS and an independent RBridge.

FIGURE 28 VCS cluster and independent RBridge for LST implementation



- Log in to the independent RBridge and enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 3/0/1
```

3. Enter the **track interface** command for the VCS principal RBridge.

```
device(conf-if-te-3/0/1)# track interface port-channel 30
```

4. To enable tracking of the VCS from the independent RBridge, enter the **track enable** command.

```
device(conf-if-te-3/0/1)# track enable
```

5. Log out of the independent RBridge.

```
device(conf-if-te-3/0/1)# end
device# exit
```

6. Log in to the VCS principal RBridge and enter **configure** to access global configuration mode.

```
device# configure
```

7. Enter the **interface** command to access the downlink interface.

```
device(config)# interface port-channel 1
```

8. For each uplink interface that you want to track, enter the **track interface** command.

```
device(config-port-channel-1)# track interface ethernet 1/0/10
device(config-port-channel-1)# track interface ethernet 2/0/11
```

9. To modify the default behavior under multiple uplinks on VCS RBridges (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(config-port-channel-1)# track min-link 1
```

In this case, the downlink vLAG member port on RBridge1 shuts down only if Uplink1 goes down. Similarly, the downlink vLAG member port on RBridge2 shuts down only if Uplink2 goes down.

10. Enter the **track enable** command to enable tracking.

```
device(config-port-channel-1)# track enable
```

Disabling LST

Use this procedure to disable link-state tracking on an interface, either partially or completely.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

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


3. To disable uplink tracking, perform one of the following:

- To remove a specific uplink from tracking, enter the **no track interface** command

```
device(conf-if-te-1/0/1)# no track interface ethernet 1/0/20
```

- To disable tracking of all uplinks from this interface, enter the **no track enable** command.

```
device(conf-if-te-1/0/1)# no track enable
```

LST show commands

There is a range of show commands that include link-state tracking (LST) information. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 18 LST Show commands in the Network OS Command Reference

Command	Description
show interface	Displays the detailed interface configuration and capabilities of all interfaces or a specific interface. This command also indicates if an interface was brought down by LST because its uplinks were down.
show running-config interface <N>gigabitethernet	Displays configuration information about one or all device interfaces of a specific capacity. Replace <N> gigabitethernet with the desired operand (for example, te gigabitethernet). This command also indicates on which downlinks LST is defined, their enablement status, and any tracked uplinks.
show track summary	Displays LST details for all interfaces on a device.

Unicast Reverse Path Forwarding (uRPF)

• uRPF overview.....	163
• Devices supported for uRPF.....	163
• uRPF configuration guidelines.....	163
• uRPF implementation.....	164
• uRPF show commands	165

uRPF overview

Unicast Reverse Path Forwarding (uRPF) prevents denial of service (DOS) attacks that use source IP spoofing and related techniques.

Common denial of service (DoS) attacks—including Smurf and Tribe Flood Network (TFN)—use forged or rapidly changing source IP addresses to thwart efforts to locate or filter the attacks. Unicast Reverse Path Forwarding (uRPF) prevents such spoofing, by verifying that the source IP address specified for a packet is from an IP address to which the device has access. IPv4 packets with invalid source IP addresses are dropped.

Network OS devices support two unicast Reverse Path Forwarding (uRPF) modes, according to RFC 3704 (“Ingress Filtering for Multihomed Networks”):

- **Loose mode:** Loose mode permits a packet if the source address matches a routing table entry. Packets are dropped only if the source address is not reachable through any device interface.
- **Strict mode:** Strict mode requires that a packet matches a known route entry—as described in loose mode—and also that it arrives at the interface in accord with router table next-hop information. Packets that do not match both of these criteria are dropped.

Both loose mode and strict mode include the default route in the Source IP (SIP) lookup.

Devices supported for uRPF

The following Network OS devices support unicast Reverse Path Forwarding (uRPF):

- VDX 6740 series
- VDX 6940 series

uRPF configuration guidelines

The following configuration guidelines apply to unicast Reverse Path Forwarding (uRPF):

- uRPF is supported on Layer 3 interfaces (physical interfaces, port-channels, and VEs).
- uRPF is VRF-aware.
- Although loose mode supports ECMP routes, strict mode does not.
- If a VLAN has multiple ports, the uRPF check does not identify packets coming in from different ports within the same VLAN, because a VLAN is considered as having a single Layer 3 interface.
- uRPF logging is not supported.

The following guidelines concern priorities and interactions among uRPF, access-control lists (ACLs), and policy-based routing (PBR):

- A packet matching an ACL deny rule is dropped irrespective of the uRPF result. Only the ACL counter is incremented; the uRPF drop counter is not incremented (even if the packet fails uRPF check).
- If a packet matches an ACL permit rule, the packet is forwarded or dropped according to the uRPF result. The ACL counter is incremented; the uRPF drop counter is incremented only if the packet fails uRPF check.
- Policy Based Routing (PBR) has precedence over uRPF. Packets are forwarded based on the PBR result, irrespective of the uRPF result.
- uRPF can be configured concurrently with routing protocol configurations and multicast configurations.

uRPF implementation

You can implement uRPF on Layer 3 physical, port-channel, and VE interfaces.

Configuring uRPF on a physical interface

Use this topic to enable or to disable unicast Reverse Path Forwarding (uRPF) on a physical interface.

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

```
device# configure terminal
```

2. Enter the **interface** command, specifying the interface type and the rbridge-id/slot/port number.

```
device(config)# interface ten 10/5/22
```

3. To enable uRPF on the interface, enter the **rpf-mode** command, specifying **loose** or **strict**.

```
device(config-if-te-10/5/22)# rpf-mode loose
```

4. To disable uRPF on the interface, enter the **no rpf-mode** command.

```
device(config-if-te-10/5/22)# no rpf-mode
```

Configuring uRPF on a port-channel interface

Use this topic to enable or to disable unicast Reverse Path Forwarding (uRPF) on a port-channel interface.

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

```
device# configure terminal
```

2. Enter the **interface port-channel** command, specifying the port-channel number.

```
device(config)# interface port-channel 10
```

3. To enable uRPF on the interface, enter the **rpf-mode** command, specifying **loose** or **strict**.

```
device(config-Port-channel-10)# rpf-mode loose
```

4. To disable uRPF on the interface, enter the **no rpf-mode** command.

```
device(config-Port-channel-10)# no rpf-mode
```

Configuring uRPF on a VE interface

Use this topic to enable or to disable unicast Reverse Path Forwarding (uRPF) on a Virtual Ethernet (VE).

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

```
device# configure terminal
```

2. Enter the **rbridge-id** command, specifying the ID.

```
device(config)# rbridge-id 2
```

3. Enter the **interface ve** command, specifying the ID.

```
device(config-rbridge-id-1)# interface ve 3
```

4. To enable uRPF on the interface, enter the **rpf-mode** command, specifying **loose** or **strict**.

```
device(config-ve-3)# rpf-mode loose
```

5. To disable uRPF on the interface, enter the **no rpf-mode** command.

```
device(config-ve-3)# no rpf-mode
```

uRPF show commands

There are several show commands that display unicast Reverse Path Forwarding (uRPF) information, listed here with descriptions.

TABLE 19 uRPF show commands in the *Command Reference*

Command	Description
show ip interface	Displays the IP interface status and configuration—including the RPF mode—of all interfaces or a specified interface.
show statistics rpf	Displays the number of packets dropped under uRPF.