Extreme SLX-OS Layer 2 Switching Configuration Guide, 20.6.2

Supporting ExtremeRouting and ExtremeSwitching SLX 9740, SLX 9640, SLX 9540, SLX 9250, SLX 9150, Extreme 8820, Extreme 8720, and Extreme 8520

9038892-00 Rev AA
June 2024

Copyright © 2024 Extreme Networks, Inc. All rights reserved.

Legal Notice

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

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

Trademarks

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

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

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

Open Source Declarations

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

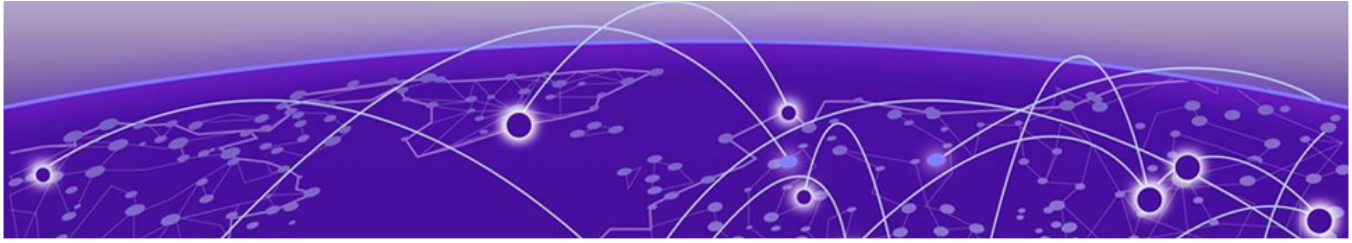


Table of Contents

Preface **11**
- Text Conventions 11
- Documentation and Training 12
  - Open Source Declarations 13
  - Training 13
- Help and Support 13
  - Subscribe to Product Announcements 14
- Send Feedback 14

About This Document **15**
- What's New in this Document 15
- Supported Hardware 15
- Regarding Ethernet interfaces and chassis devices 16

Link Aggregation **17**
- Link aggregation overview 17
  - LAG configuration guidelines 18
  - LAG profile support-scope 19
- Basic LAG configuration 20
  - Configuring a new port channel interface 20
  - Deleting a port channel interface 21
  - Adding a member port to a port channel 21
  - Deleting a member port from a port channel 22
  - Configuring the minimum number of LAG member links 22
- Dynamic (LACP) configuration 23
  - Configuring LACP system priority 24
  - Configuring the LACP port priority 24
  - Configuring the LACP timeout period 25
  - LACP PDU forwarding 25
  - Configuring LACP default Up 26
- IP over port-channel 26
  - Commands supported for IP over port-channel 27
  - IP over port-channel limitations 34
- Hash-based Load Balancing 34
  - Configuring LAG Hashing 34
  - Configuring header protocols for load-balancing 36
  - Load balancing mechanism on different traffic types 37
  - Load balance hash support on different devices 38
  - LAG hash support on different devices 40
  - MPLS transit load balancing 40
- Troubleshooting LAGs 42
  - Troubleshooting static LAGs 42

Troubleshooting dynamic (LACP) LAGs....42
Show and clear LAG commands....43
Displaying port-channel information....43
Displaying LAG hashing....44
Displaying LACP system-id information....44
Displaying LACP statistics....44
Clearing LACP counter statistics on a LAG....45
Clearing LACP counter statistics on all LAG groups....45

VLANs....46
802.1Q VLAN overview....46
Configuring VLANs....46
Configuring a VLAN....46
Configuring a switchport interface....47
Configuring the switchport interface mode....47
Configuring the switchport access VLAN type....48
Configuring a VLAN in trunk mode....48
Configuring a native VLAN on a trunk port....49
Enabling VLAN tagging for native traffic....49
Displaying the status of a switchport interface....50
Displaying the switchport interface type....51
Verifying a switchport interface running configuration....51
Displaying VLAN information....52
Enabling Layer 3 routing for VLANs....52
VLAN statistics....53
Enabling statistics on a VLAN....53
Displaying statistics for VLANs....54
Clearing statistics on VLANs....54
VE route-only mode....55
Configuring VE route-only mode on a physical port....55
Configuring VE route-only mode on a LAG port....56

VxLAN Layer 2 Gateway....58
VxLAN Layer 2 gateway overview....58
VxLAN Layer 2 gateway considerations and limitations....60
VNI Mapping....61
Configuring VxLAN Layer 2 gateway....61
VxLAN Layer 2 gateway support for bridge domains....63
Configuring VxLAN Layer 2 Gateway support for bridge domains....64
VxLAN Layer 2 gateway payload tag processing....65
VxLAN Layer 2 support for LVTEP....66
LVTEP control plane....66
LVTEP data plane....66
Port-based VLAN bundle service....68
Configuring VxLAN LVTEP support....69
LVTEP support for other features....72
Nondefault TPID....75
Configuring TCAM profiles to support LVTEP....77
LVTEP show commands....78

QoS for VxLAN Layer 2 gateways....81

Configuring QoS for VxLAN Layer 2 gateways....82

Multiple VLAN Registration Protocol (MVRP)....84

Multiple VLAN Registration Protocol overview....84
MVRP considerations and limitations....85
Enabling MVRP on an Ethernet interface....86
Configuring an edge port....87
Configuring the VLAN registration mode on the interface....88
MVRP over a port channel....88
Enabling MVRP over a port channel....88
Configuring the MVRP join, leave, and leave-all timers....90
Displaying MVRP configuration information, statistics, and attributes....91
Clearing MVRP statistics....93

Multi-Chassis Trunking (MCT)....94

MCT Overview....94
MCT terminology....95
SLX-OS MCT control plane....95
MCT Data Plane for the SLX Devices....96
MAC management....102
MCT configuration considerations....106
General considerations....106
Peer considerations....106
VLAN considerations....106
LACP considerations....106
Configuring MCT....107
Taking the MCT node offline for maintenance....108
Configuring additional MCT cluster parameters....109
Peer Keepalive....109
Configuring peer-keepalive destination....110
Moving the traffic from an MCT node to the remote node....110
Displaying MCT information....110
Displaying the cluster information....110
Displaying the cluster client information....111
Displaying member VLAN information....112
Displaying and clearing the MAC address table cluster information....112
VPLS and VLL MCT on the SLX 9640 and 9540 devices....112
Control plane for VPLS or VLL MCT....113
PW state in VPLS or VLL MCT....113
VLL-MCT data plane....114
VPLS-MCT data plane....115
VPLS MAC Management....118
Configuration Considerations and Limitations for VPLS and VLL MCT....121
Configuring MCT for VPLS or VLL....121
Displaying information related to VPLS and VLL MCT....122
Layer 3 routing over MCT....124
Configuration considerations....125
Layer 3 MCT VLAN configuration example....126
Layer 3 MCT bridge-domain configuration example....127
Using MCT with VRRP and VRRP-E....128

MCT short path forwarding configuration using VRRP-E example ... 129
PE1 configuration ... 130
PE2 configuration ... 130
MCT Use Cases ... 131
L2 MCT in the data center core ... 131
L2 MCT in a data center with a collapsed core and aggregation ... 133

Logical Interfaces ... 135
Logical interfaces overview ... 135
LIFs and bridge domains ... 135
Configuration considerations ... 135
Configuring a logical interface on a physical port or port-channel (LAG) ... 136

Bridge Domains ... 138
Bridge domain overview ... 138
Bridge domain statistics ... 138
Configuring a bridge domain ... 139
Displaying bridge-domain configuration information ... 140
Enabling statistics on a bridge domain ... 144
Displaying statistics for logical interfaces in bridge domains ... 144
Clearing statistics on bridge domains ... 145

VPLS and VLL Layer 2 VPN services ... 146
VPLS overview ... 146
VLL ... 149
VPLS service endpoints ... 150
Pseudowires ... 151
Supported VPLS features ... 157
Configuration of VPLS and VLL ... 158
QoS treatment in VPLS packet flow ... 158
Configuring a PW profile ... 159
Attaching a PW profile to a bridge domain ... 160
Configuring control word for a PW profile ... 160
Configuring PW control word on a bridge domain ... 161
Configuring flow label for a PW profile ... 162
Configuring PW flow label on a bridge domain ... 163
Configuring a static MAC address over an endpoint in a VPLS instance ... 164
Displaying MAC address information for VPLS bridge domains ... 165
Configuring a VPLS instance ... 165
Configuring a VLL instance ... 167
Routing VE over VPLS ... 168
Configuration example for VPLS with switching between ACs and network core ... 171
PE1 ... 171
PE2 ... 172
VPLS MAC withdrawal ... 172
Enabling VPLS MAC address withdrawal ... 172

MAC Movement Detection and Resolution ... 174
MAC Movement Overview ... 174
MAC Movement Detection ... 175
MAC Movement Resolution ... 175
MAC movement Detection and Resolution Commands ... 177

802.1ag Connectivity Fault Management....179
Maintenance Domain (MD)....180
Maintenance Association (MA)....181
Maintenance End Point (MEP)....181
Maintenance Intermediate Point (MIP)....181
CFM Hierarchy....182
Mechanisms of Ethernet IEEE 802.1ag OAM....182
Fault detection (continuity check message)....182
Fault verification (Loopback messages)....182
Fault isolation (Linktrace messages)....182
Enabling or disabling CFM....183
Creating a Maintenance Domain....183
Creating and configuring a Maintenance Association....183
Displaying CFM configurations....184
show cfm....184
show cfm connectivity....185
show cfm brief....185

802.1d Spanning Tree Protocol....187
Spanning Tree Protocol overview....187
Spanning Tree Protocol configuration notes....187
Optional features....188
STP states....188
BPDUs....188
TCN BPDUs....189
STP configuration guidelines and restrictions....190
Understanding the default STP configuration....190
STP features....191
Root guard....191
BPDU guard....191
Error disable recovery....192
PortFast....192
STP parameters....192
Bridge parameters....193
Error disable timeout parameter....194
Port-channel path cost parameter....194
Configuring STP....195
Enabling and configuring STP globally....195
Enabling and configuring STP on an interface....197
Configuring basic STP parameters....199
Re-enabling an error-disabled port automatically....202
Clearing spanning tree counters....202
Clearing spanning tree-detected protocols....203
Shutting down STP....203

802.1w Rapid Spanning Tree Protocol....205
Rapid Spanning Tree Protocol overview....205
RSTP Parameters....206
Edge port and automatic edge detection....206
Configuring RSTP....207

Enabling and configuring RSTP globally 207
Enabling and configuring RSTP on an interface 209
Configuring basic RSTP parameters 211
Clearing spanning tree counters 213
Clearing spanning tree-detected protocols 214
Shutting down RSTP 214

Per-VLAN Spanning Tree+ and Rapid Per-VLAN Spanning Tree+ 215
PVST+ and R-PVST+ overview 215
PVST+ and R-PVST+ guidelines and restrictions 215
PVST+ and R-PVST+ parameters 216
Bridge protocol data units in different VLANs 216
BPDU configuration notes 217
PortFast 221
Edge port and automatic edge detection 221
Configuring PVST+ and R-PVST+ 222
Enabling and configuring PVST+ globally 222
Enabling and configuring PVST+ on an interface 224
Enabling and configuring PVST+ on a system 226
Enabling and configuring R-PVST+ globally 232
Enabling and configuring R-PVST+ on an interface 233
Enabling and configuring R-PVST+ on a system 235
Clearing spanning tree counters 242
Clearing spanning tree-detected protocols 242
Shutting down PVST+ or R-PVST+ 242

802.1s Multiple Spanning Tree Protocol 244
MSTP overview 244
Common Spanning Tree (CST) 244
Internal Spanning Tree (IST) 245
Common Internal Spanning Tree (CIST) 245
Multiple Spanning Tree Instance (MSTI) 245
MST regions 245
MSTP regions 245
MSTP guidelines and restrictions 246
Interoperability with PVST+ and R-PVST+ 247
MSTP global level parameters 247
MSTP interface level parameters 247
Edge port and automatic edge detection 247
BPDU guard 248
Restricted role 249
Restricted TCN 249
Configuring MSTP 249
Enabling and configuring MSTP globally 249
Enabling and configuring MSTP on an interface 253
Enabling MSTP on a VLAN 255
Configuring basic MSTP parameters 256
Clearing spanning tree counters 258
Clearing spanning tree-detected protocols 258
Shutting down MSTP 258

Topology Groups **260**
Topology groups 260
Master VLAN, member VLANs, and bridge-domains 260
Control ports and free ports 261
Configuration considerations 261
Configuring a topology group 262
Configuring a master VLAN 262
Adding member VLANs 262
Adding member bridge-domains 263
Replacing a master VLAN 264
Displaying topology group information 264

Loop Detection **266**
LD protocol overview 266
Strict mode 266
Loose mode 267
LD PDU format 268
LD PDU transmission 269
LD PDU reception 269
LD parameters 270
LD PDU processing 271
Support for EPVN VLAN tunnels 272
Configuration considerations 272
LD use cases 272
MCT strict mode 272
MCT loose mode 273
Configuring LD protocol 275
Loop detection for VLAN 277
Configuring loop detection for VLAN 277

Ethernet Ring Protection Protocol **279**
Ethernet Ring Protection overview 279
Configuration Considerations 279
ERP components 280
Initializing a new ERN 284
Signal Fail 289
Manual Switch 290
Forced Switch 293
Double Forced Switch 297
Dual-end blocking 297
Non-revertive mode 298
Interconnected rings 298
Configuring ERP 299
ERP topology and configuration 300

ETH-CSF **313**
ETH-CSF overview 313
ETH-CSF use case 313
ETH-CSF specifications 315
ETH-CSF and port-channel 315
CSF transmission 316

CSF reception 316
ETH-CSF PDU structure 317
ETH-CSF considerations and limitations 318
Configuring ETH-CSF 319

Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as Extreme Networks switches or SLX routers, the product is referred to as *the switch* or *the router*.

Table 1: Notes and warnings

Icon	Notice type	Alerts you to...
	Tip	Helpful tips and notices for using the product
	Note	Useful information or instructions
	Important	Important features or instructions
	Caution	Risk of personal injury, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 2: Text

Convention	Description
`screen displays`	This typeface indicates command syntax, or represents information as it is displayed on the screen.
The words *enter* and *type*	When you see the word *enter* in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says *type*.
Key names	Key names are written in boldface, for example **Ctrl** or **Esc**. If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press **Ctrl**+**Alt**+**Del**
Words in italicized type	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
italic text	Italic text indicates variable content.
`[ ]`	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, such as passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	In command examples, the backslash indicates a “soft” line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

Current Product Documentation
Release Notes
Hardware and Software Compatibility for Extreme Networks products
Extreme Optics Compatibility
Other Resources such as articles, white papers, and case studies

Open Source Declarations

Some software files have been licensed under certain open source licenses. Information is available on the Open Source Declaration page.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit the Extreme Networks Training page.

Help and Support

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

Extreme Portal

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

The Hub

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

Call GTAC

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

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

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

Subscribe to Product Announcements

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

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

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

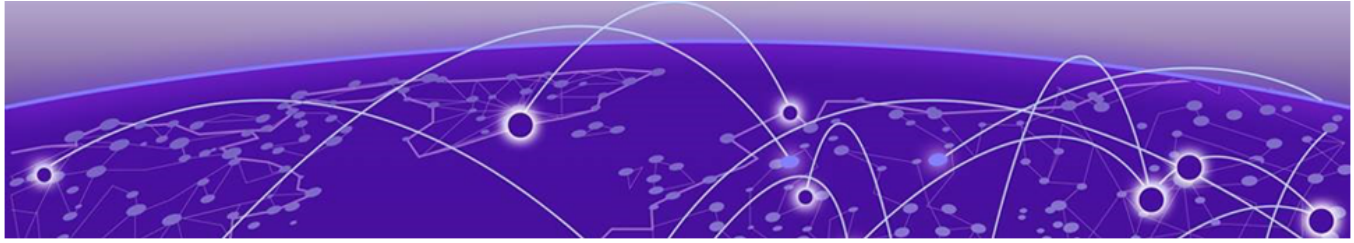
Send Feedback

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

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

To send feedback, email us at documentation@extremenetworks.com.

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



About This Document

What's New in this Document on page 15
Supported Hardware on page 15
Regarding Ethernet interfaces and chassis devices on page 16

What's New in this Document

This document is released with the SLX-OS 20.6.2 software release.

The following table includes descriptions of changes made to this document for the SLX-OS 20.6.2 software release.

Table 4: Changes for the current release

Feature	Description	Described in
Hash-based Load Balance support for SLX 9740 and Extreme 8820.	The support for Load balance hash and LAG hash was implemented in the SLX 9740 and Extreme 8820 devices.	Hash-based load balancing in the chapter Link Aggregation .

For additional information, refer to the *Extreme SLX-OS Release Notes* for this version.

Supported Hardware

SLX-OS 20.6.2 supports the following hardware platforms.

- Extreme 8820
- Extreme 8720
- Extreme 8520
- ExtremeSwitching SLX 9540
- ExtremeSwitching SLX 9250
- ExtremeSwitching SLX 9150

- ExtremeRouting SLX 9740
- ExtremeRouting SLX 9640

Note

All configurations and software features that are applicable to SLX 9150 and SLX 9250 devices are also applicable for the Extreme 8520 and Extreme 8720 devices respectively.

All configurations and software features that are applicable to SLX 9740 devices are also applicable for the Extreme 8820 devices.

The "Measured Boot with Remote Attestation" feature is only applicable to the Extreme 8520, Extreme 8720, and Extreme 8820 devices. It is not supported on the SLX 9150 and SLX 9250 devices.

Note

Although many software and hardware configurations are tested and supported for this release, documenting all possible configurations and scenarios is beyond this document's scope.

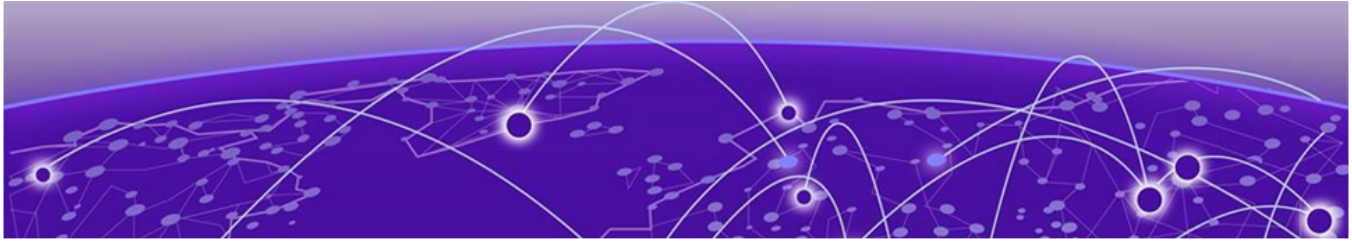
For information about other releases, see the documentation for those releases.

Regarding Ethernet interfaces and chassis devices

The current SLX-OS version does not support any multi-slot (chassis) devices.

However, the Ethernet interface configuration and output *slot/port* examples in this document may appear as either 0/x or n/x, where "n" and "x" are integers greater than 0.

For all currently supported devices, specify **0** for the slot number.



Link Aggregation

Link aggregation overview on page 17
Basic LAG configuration on page 20
Dynamic (LACP) configuration on page 23
IP over port-channel on page 26
Hash-based Load Balancing on page 34
Troubleshooting LAGs on page 42
Show and clear LAG commands on page 43

Link aggregation overview

We also refer to port-channels as link-aggregation groups (LAGs). A LAG is considered 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 one link fails.

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 port-channel.

The benefits of link aggregation are as follows:

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

Each LAG consists of the following components:

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

Two LAG types are supported:

- Static LAGs—In static link aggregation, links are added into a LAG without exchanging 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 LAGs—Dynamic link aggregation uses Link Aggregation Control Protocol (LACP) to negotiate the links included in 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 LACP protocol data units (PDUs) to monitor the health of each member link.

LAG configuration guidelines

- You can associate a link with only one port-channel.
- You cannot aggregate switchport interfaces into port-channels. However, you can configure port-channels as switchports.

LAG profile support-scope

SLX 9740/Extreme 8820 devices

Port channel scale and support for SLX 9740/Extreme 8820.

Table 5: Port-channel scale for SLX 9740/Extreme 8820 devices.

Device	LAG Profile	Supported port-channel IDs	Maximum links per port-channel
SLX 9740-40/ Extreme 8820-40	default	1-256; Only 77 portchannels may be created at any one time.	64
SLX 9740-80/ Extreme 8820-80	default	1-256; Only 153 port-channels may be created at one time.	64

Note

- For the 1U SLX 9740-40/Extreme 8820-40, the number of LAGs will be 77, where:
  - 76 are the front end ports (all breakouts)
  - 1 (insight port)
- For the 2U SLX 9740-80/Extreme 8820-80, the number of LAGs will be 153. where:
  - 152 are the front end ports (all breakouts)
  - 1 (insight port)

SLX 9540 and SLX 9640 devices

For this group of devices, maximum numbers of port-channel IDs and links per port-channel vary with device and LAG profile, as follows:

Table 6: Port-channel scale for SLX 9540 and SLX 9640 devices

Device or series	LAG profile	Supported port-channel IDs	Maximum links per port-channel
SLX 9540 SLX 9640	default	1–256; Only 64 port-channels may be created at any one time.	64
SLX 9540 SLX 9640	lag-profile-1	1–256; Only 64 port-channels may be created at any one time.	32

SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices

For this group of devices, maximum numbers of port-channel IDs and links per port-channel vary only with device, as follows:

Table 7: Port-channel support for SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices

Device or series	Supported port-channel IDs	Maximum links per port-channel
SLX 9150 SLX 9250 Extreme 8520 Extreme 8720	1–256; Only 128 port-channels may be created at any one time.	64

Note
Non-default LAG profiles are not supported for the SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.

Example

The following example enables **lag-profile-1**.

```
device# configure terminal
device(config)# hardware
device(config-hardware)# profile lag lag-profile-1
%Warning: To activate the new profile config, please run 'copy running-config startup-
config' followed by 'reload system'.
device(config-hardware)#
```

Basic LAG configuration

The topics in this section configure both static and dynamic (LACP) LAG implementations.

Configuring a new port channel interface

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

```
device# configure terminal
```

2. Enter the **interface port-channel** command to create a new port channel interface.

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

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

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

After creating a new port channel, you can enter **no shutdown** or **shutdown** to bring up or down the port-channel, as follows:

```
device# configuration terminal
device(config)# interface Port-channel 30
2016/10/17-20:31:21, [NSM-1004], 302, M2 | Active | DCE, INFO, SLX,  Port-channel 30 is
created.
device(config-Port-channel-30)#
device(config-Port-channel-30)# no shutdown
2016/10/17-20:31:26, [NSM-1019], 303, M2 | Active | DCE, INFO, SLX,  Interface Port-
channel 30 is administratively up.
device(config-Port-channel-30)#
```

Deleting a port channel interface

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

```
device# configure terminal
```

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

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

The following example deletes port-channel interface 30.

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

Adding a member port to a port channel

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note

Run the **no shutdown** command to bring the above interface online.

```
device(conf-if-eth-0/5)# no shutdown
2016/10/18-03:47:15, [NSM-1019], 528, M2 | Active | DCE, INFO, SLX,  Interface
Ethernet 0/5 is administratively up. 2016/10/18-03:47:15, [NSM-1001], 529, M2 |
Active | DCE, INFO, SLX,  Interface Ethernet 0/5 is online.
```

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

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

Deleting a member port from a port channel

Delete a port from the port channel interface.

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

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

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

Configuring the minimum number of LAG member links

This configuration allows a port-channel to operate at a certain minimum bandwidth at all times. If the bandwidth of the port-channel drops below the minimum number,

then the port-channel is declared operationally DOWN even though it has operationally UP members.

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

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

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

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

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

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

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

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

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

Dynamic (LACP) configuration

If LACP determines that 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 protocol data unit (PDU) exchanges, regardless of whether the partner system sends LACP PDUs.
- *Passive mode*—LACP responds to PDUs initiated by its partner system, but does not initiate the LACP PDU exchange.

The LACP process collects and distributes Ethernet frames. The collection and distribution process implements:

- Inserting and capturing control LACP protocol data units (PDUs).
- Restricting the traffic of a given conversation to a specific link.
- Load-balancing links.
- Handling dynamic changes in LAG membership.

On each port, link aggregation control:

- 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.

Configuring LACP system priority

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.

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

```
device# configure terminal
```

2. Specify the LACP system priority.

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

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

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

Configuring the LACP port priority

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

```
device# configure terminal
```

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

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

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

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

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

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

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

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

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

The example sets the port priority as 12.

```
device# configure terminal
```

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

Configuring the LACP timeout period

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

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

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

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

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

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

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

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

LACP PDU forwarding

Since the destination address of the PDU is a multicast MAC, the frame will be flooded on the VLAN. If the VLAN on which the LACP PDU is received is a regular VLAN, the PDU will be flooded on the VLAN. If the VLAN on which the PDU is received is a service delimiter for a bridge domain, the LACP PDU is flooded on the bridge domain accordingly.

LACP PDU forwarding is supported only on physical interfaces and static port channel interfaces. LACP PDUs cannot be forwarded if they are received on a LACP based dynamic port channel. LACP PDU forwarding enabled on a static port channel applies to all the member ports. If LACP is enabled on a port, it overrides the LACP PDU forwarding configuration and the PDUs are trapped in the CPU.

Configuring LACP PDU forwarding on a physical interface

1. Enter the global configuration mode.

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

2. Specify the physical interface on which LACP PDU forwarding needs to enabled.

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

3. Configure LACP PDU forwarding on the physical interface.

```
device(conf-if-eth-0/1)# lacp-pdu-forward enable
```

The following example enables LACP forwarding on a port-channel interface.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# lacp-pdu-forward enable
```

Configuring LACP PDU forwarding on a port-channel interface

1. Enter the global configuration mode.

   ```
   device# configure terminal
   ```

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

   ```
   device(config)# interface port-channel 10
   ```

3. Configure LACP PDU forwarding on the port-channel interface.

   ```
   device(conf-Port-channel-10)# lacp-pdu-forward enable
   ```

   LACP PDU forwarding is supported only on static port channel interfaces.

The following example enables LACP forwarding on a port-channel interface.

```
device# configure terminal
device(config)# interface port-channel 10
device(conf-Port-channel-10)# lacp-pdu-forward enable
```

Configuring LACP default Up

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

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

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

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

3. Specify LACP default-up for the interface.

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

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

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

IP over port-channel

IP over port-channel:

- Provides fault tolerance for the links. Protocols associated with the IP are not forced to reconverge unless all of the links in the port-channel go down.
- 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.)

IP over port-channel supports:

- Static and dynamic port-channels
- IPv4 and IPv6 addressing
- Configuration and show commands

The following Layer 3 protocols are supported:

- ACLs
- ARP/ND
- BFD
- BGP
- DHCP
- ICMP
- ISIS
- MPLS
- OSPF v2/v3
- VRRP

Commands supported for IP over port-channel

The following commands support IP over port-channel:

Table 8: IP routing commands for IP over port-channel

Command	Mode	Support limitations
ip policy route-map	Port-channel configuration	Not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.
ip route	Global configuration	
ipv6 policy route-map	Port-channel configuration	Not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.
ipv6 route	Global configuration	
show route-map interface port-channel	Privileged EXEC	

Table 9: Interface commands for IP over port-channel

Command	Mode	Support limitations
clear ipv6 counters interface port-channel	Privileged EXEC	
ip address *ip address* { **secondary | ospf-passive | ospf-ignore** }	Port-channel configuration	
ip directed-broadcast	Port-channel configuration	

Table 9: Interface commands for IP over port-channel (continued)

Command	Mode	Support limitations
ip mtu	Port-channel configuration	
ip unnumbered	Port-channel configuration	
ipv6 address	Port-channel configuration	
ipv6 address secondary	Port-channel configuration	
ipv6 address use-link-local-only	Port-channel configuration	
ipv6 address eui-64	Port-channel configuration	
ipv6 address eui-64 secondary	Port-channel configuration	
ipv6 address link-local	Port-channel configuration	
ipv6 address anycast	Port-channel configuration	
show ip interface port-channel	Privileged EXEC	
show ipv6 interface port-channel	Privileged EXEC	
show ipv6 counters interface port-channel	Privileged EXEC	
vrf forwarding	Port-channel configuration	

Table 10: ACL commands for IP over port-channel

Command	Mode	Support limitations
ip access-group	Port-channel configuration	Both ingress and egress are supported.
ip-subnet-broadcast-acl	Port-channel configuration	Not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.
ip receive access-group	Global configuration	
ipv6 access-group	Port-channel configuration	Only ingress is supported.
ipv6 receive access-group	Global configuration	
service-policy in	Port-channel configuration	

Table 11: ARP/ND commands for IP over port-channel

Command	Mode	Support limitations
arp interface port-channel	Global configuration mode	
ip proxy-arp	Port-channel configuration	

Table 11: ARP/ND commands for IP over port-channel (continued)

Command	Mode	Support limitations
ip arp-aging-timeout	Port-channel configuration	
ip arp learn-any	Port-channel configuration	
ipv6 nd broadcast-mac	Port-channel configuration	
ipv6 nd broadcast-mac-trap	Port-channel configuration	
ipv6 nd cache	Port-channel configuration	
ipv6 nd dad	Port-channel configuration	
ipv6 nd hoplimit	Port-channel configuration	
ipv6 nd managed-config-flag	Port-channel configuration	
ipv6 nd messages	Port-channel configuration	
ipv6 nd mtu	Port-channel configuration	
ipv6 nd ns-interval	Port-channel configuration	
ipv6 nd other-config-flag	Port-channel configuration	
ipv6 nd prefix	Port-channel configuration	
ipv6 nd ra-interval	Port-channel configuration	
ipv6 nd ra-lifetime	Port-channel configuration	
ipv6 nd reachable-time	Port-channel configuration	
ipv6 nd refreshed	Port-channel configuration	
ipv6 nd retrans-timer	Port-channel configuration	
ipv6 nd suppress-ra	Port-channel configuration	

Table 12: BFD commands for IP over port-channel

Command	Mode	Support limitations
bfd interval	Port-channel configuration	
bfd shutdown	Port-channel configuration	

Table 12: BFD commands for IP over port-channel (continued)

Command	Mode	Support limitations
show bfd neighbors dest-ip	Privileged EXEC	
show bfd neighbors interface port-channel	Privileged EXEC	

Table 13: BGP commands for IP over port-channel

Command	Mode	Support limitations
distribute	BGP configuration	Not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.
neighbor { *ip-address* | *ipv6-address* | *peer-group-name* } **update-source port-channel**	BGP configuration	

Table 14: DHCP commands for IP over port-channel

Command	Mode	Support limitations
ip dhcp relay address	Port-channel configuration	
ip dhcp relay gateway address	Port-channel configuration	
ipv6 dhcp relay address	Port-channel configuration	
show ip dhcp relay address interface port-channel	Privileged EXEC	
show ip dhcp relay gateway interface port-channel	Privileged EXEC	

Table 15: ICMP commands for IP over port-channel

Command	Mode	Support limitations
ip icmp address-mask	Port-channel configuration	
ip icmp echo-reply	Port-channel configuration	
ip icmp rate-limiting	Port-channel configuration	

Table 15: ICMP commands for IP over port-channel (continued)

Command	Mode	Support limitations
ip icmp redirect	Port-channel configuration	
ip icmp unreachable	Port-channel configuration	

Table 16: ISIS commands for IP over port-channel

Command	Mode	Support limitations
isis auth-check	Port-channel configuration	
isis auth-key	Port-channel configuration	
isis auth-mode md5	Port-channel configuration	
isis bfd	Port-channel configuration	
isis hello-multiplier	Port-channel configuration	
isis hello-interval	Port-channel configuration	
isis hello padding disable	Port-channel configuration	
isis ipv6 metric	Port-channel configuration	
isis ldp-sync	Port-channel configuration	
isis metric	Port-channel configuration	
isis passive	Port-channel configuration	
isis point-to-point	Port-channel configuration	
isis priority	Port-channel configuration	
isis reverse-metric	Port-channel configuration	
show isis interface port-channel	Port-channel configuration	

Note: MPLS commands are not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.

Table 17: MPLS commands for IP over port-channel

Command	Mode	Support limitations
exclude-interface port-channel	MPLS router Bypass LSP configuration	
mpls-interface port-channel	MPLS configuration	
show mpls dynamic bypass port-channel	Privileged EXEC	
show mpls interface port-channel	Privileged EXEC	

Table 17: MPLS commands for IP over port-channel (continued)

Command	Mode	Support limitations
show mpls ldp interface port-channel	Privileged EXEC	
show mpls rsvp	Privileged EXEC	

Table 18: OSPFv2 commands for IP over port-channel

Command	Mode	Support limitations
clear ip ospf counters port-channel	Privileged EXEC	
ip ospf active	Port-channel configuration	
ip ospf area	Port-channel configuration	
ip ospf auth-change-wait-time	Port-channel configuration	
ip ospf authentication-key	Port-channel configuration	
ip ospf bfd	Port-channel configuration	
ip ospf cost	Port-channel configuration	
ip ospf database-filter	Port-channel configuration	
ip ospf dead-interval	Port-channel configuration	
ip ospf hello-interval	Port-channel configuration	
ip ospf md5-authentication	Port-channel configuration	
ip ospf mtu-ignore	Port-channel configuration	
ip ospf network	Port-channel configuration	
ip ospf passive	Port-channel configuration	
ip ospf priority	Port-channel configuration	
ip ospf retransmit-interval	Port-channel configuration	
ip ospf transmit-delay	Port-channel configuration	
show debug ip ospf internal interface port-channel	Privileged EXEC	

Table 18: OSPFv2 commands for IP over port-channel (continued)

Command	Mode	Support limitations
show ip ospf interface port-channel	Privileged EXEC	
show ip ospf neighbor port-channel	Privileged EXEC	

Table 19: OSPFv3 commands for IP over port-channel

Command	Mode	Support limitations
clear ipv6 ospf counts neighbor interface port-channel	Privileged EXEC	
clear ipv6 ospf neighbor interface port-channel	Privileged EXEC	
ipv6 ospf active	Port-channel configuration	
ipv6 ospf area	Port-channel configuration	
ipv6 ospf autentication	Port-channel configuration	
ipv6 ospf bfd	Port-channel configuration	
ipv6 ospf cost	Port-channel configuration	
ipv6 ospf dead-interval	Port-channel configuration	
ipv6 ospf hello-interval	Port-channel configuration	
ipv6 ospf hello-jitter	Port-channel configuration	
ipv6 ospf instance	Port-channel configuration	
ipv6 ospf mtu-ignore	Port-channel configuration	
ipv6 ospf network	Port-channel configuration	
ipv6 ospf passive	Port-channel configuration	
ipv6 ospf priority	Port-channel configuration	
ipv6 ospf retransmit-interval	Port-channel configuration	
ipv6 ospf suppress-linklsa	Port-channel configuration	
ipv6 ospf transmit-delay	Port-channel configuration	

Table 19: OSPFv3 commands for IP over port-channel (continued)

Command	Mode	Support limitations
show ipv6 ospf interface port-channel	Privileged EXEC	
show ipv6 ospf neighbor interface port-channel	Privileged EXEC	

Table 20: VRRP commands for IP over port-channel

Command	Mode	Support limitations
ipv6 vrrp-group	Port-channel configuration	
vrrp-group	Port-channel configuration	
vrrp-extended-group	Port-channel configuration	
show vrrp interface port-channel	Privileged EXEC	
show ipv6 vrrp interface port-channel	Privileged EXEC	

IP over port-channel limitations

Note the following limitations:

- IP on MCT client port-channel is not supported.
- Port-channels are not supported as source interfaces for manageability clients like SSH, sFlow, Radius, TACACS+, or Syslog.

Hash-based Load Balancing

Hash-based load balancing shares traffic load across LAG ports, while ensuring that packets in the flow are not reordered.

Note
The MPLS options in this section are not supported on the SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.

Configuring LAG Hashing

To configure symmetric LAG hashing on supported devices, complete the following tasks.

1. Define where to start picking headers for the key generation, using the **lag hash hdr-start** command.
   - **fwd** - Start from the header that is used for the forwarding of the packet (inner header). This is the default option.

- **term** - Start from the last terminated header (outer header) - the header after the forwarding header. For switching traffic, as there is no header below the forwarding header, hashing is not visible.

2. Configure the number of headers to be considered for LAG hashing, using the **lag hash hdr-count** command. The default value is 1. There can be a maximum of 3 headers - based on the first header selected using the command in the previous step.

The following options provide other LAG configurations to achieve specific tasks:

- Configure hash rotate using the **lag hash rotate** command to provide different options for randomness of hashing. The number can be between 0 and 15. The default value is 3.
- If there is a need to use the same hash in both directions, configure hash as normalize, using the **lag hash normalize** command. The normalize option is disabled by default.
- Allow the source port to be included in the hashing configuration using the **lag hash srcport** command. The source port is not used for hashing by default.
- To skip the entire MPLS label stack and pick only the BOS label for hashing, use the **lag hash bos**. By default, if the MPLS header is used for hashing, all labels, including *BOS*, are also used for hashing.
  - **start** - start from BOS. This is the default option.
  - **skip** - hash from header next to the BOS label.
- Enter the **lag hash pwctrlword** command to skip the password control word in the hashing configuration.
- The following MPLS transit node LSR hashing configuration options are available when using the **lag hash speculate-mpls** command. The default option is using the MPLS labels.
  - enable - Enables Speculative MPLS.
  - inner-eth - Enables inner ethernet header hash for L2VPN.
  - inner-ip-raw - Enables inner IPv4 header hash for L2VPN raw mode.
  - inner-ip-tag - Enables inner IPv4 header hash for L2VPN tag mode.
  - inner-ipv6-raw - Enables inner IPv6 header hash for L2VPN raw mode.
  - inner-ipv6-tag - Enables inner IPv6 header hash for L2VPN tag mode.
- Enter the **lag hash rotate** command to specify hashing randomness. A randomness value ranges from 0 to 15 with a default of 3.

Configuring header protocols for load-balancing

Select the protocol header type using one of the following commands. By default, all the header parameters are enabled as shown here. If you disable a header, you can then re-enable its parameters one-by-one.

- Ethernet headers:
  - **load-balance hash ethernet da-mac**
  - **load-balance hash ethernet etype**
  - **load-balance hash ethernet sa-mac**
  - **load-balance hash ethernet vlan**
  - **load-balance hash ethernet smac-dmac**
- IPv4 and L4 headers
  - **load-balance hash ip dst-ip**
  - **load-balance hash ip dst-l4-port**
  - **load-balance hash ip protocol**
  - **load-balance hash ip src-ip**
  - **load-balance hash ip src-l4-port**
  - **load-balance hash ip sip-dip**
- IPv6 and L4 headers
  - **load-balance hash ipv6 ipv6-dst-ip**
  - **load-balance hash ipv6 ipv6-dst-l4-port**
  - **load-balance hash ipv6 ipv6-next-hdr**
  - **load-balance hash ipv6 ipv6-src-ip**
  - **load-balance hash ipv6 ipv6-src-l4-port**
  - **load-balance hash ipv6 sipv6-dipv6**
- MPLS:
  - **load-balance hash mpls label1**
  - **load-balance hash mpls label2**
  - **load-balance hash mpls label3**
  - **load-balance hash mpls all-labels**
- L4 headers:
  - **load-balance hash l4 src-dst-port**

Load balancing mechanism on different traffic types

The following table provides information about load balancing on different traffic types.

Table 21: Load balancing on different traffic types

Traffic type	Header field	Description
Layer 2/ Layer 3 packet load balancing	• Ethernet DA, SA, Etype, Vlan-id • IPv4/v6 dst IP, src IP • L4 Src-Port, Dst-Port	• Ethernet destination address, source address, ethernet type, VLAN ID load balancing • IPv4/v6 destination address, source address load balancing • Layer 4 source and destination port-based load balancing
VPLS/ VLL packet load balancing	CE to PE router traffic can use the following fields for load-balancing similar to the Layer 2/ Layer 3 traffic) • Ethernet DA, SA, Etype, Vlan-id • IPv4/v6 dst IP, src IP • L4 Src-Port, Dst-Port PE to CE router traffic can use the following fields for load-balancing • Customer (inner) ethernet DA, SA, Etype, Vlan-id • Customer (inner) IPv4/v6 dst IP, Ipv4/ Ipv6 src IP, protocol • Customer (inner) L4 Src-Port, Dst-Port	CE to PE router traffic • Ethernet destination address, source address, ethernet type, VLAN ID load balancing • IPv4/v6 destination address, source address load balancing • Layer 4 source and destination port-based load balancing PE to CE router traffic • Customer ethernet destination and source address, ethernet type, VLAN ID load balancing • Customer IPv4/v6 destination address, source address load balancing • Customer Layer 4 source and destination port-based load balancing
MPLS LSR load balancing • Hashing options support different MPLS transit hashing scenarios • The hashing options are mutually exclusive. If one option is enabled, the other option will be disabled.	IP over MPLS traffic going over transit node	Extreme supports speculate-mpls option as default which speculates the IPv4/IPv6 header after the MPLS labels and use the fields for hashing. This hashing scenario is handled by the **`lag hash speculate-mpls enable`** command in the global mode.

Table 21: Load balancing on different traffic types (continued)

Traffic type	Header field	Description
L2VPN (VPLS/ VLL) traffic • The hashing options are mutually exclusive. If one option is enabled, the other option will be disabled.	L2VPN tagged mode with IPv4 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ip-tag** command in the global mode. Some sections of the IPv4 source and destination address fields are also used for load-balance hashing.
	L2VPN raw mode with IPv4 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ip-raw** command. Some sections of the IPv4 source and destination address fields are also used for load-balance hashing.
	L2VPN tagged mode with IPv6 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ipv6-tag** command. Some sections of the IPv6 source and destination address fields are also used for load-balance hashing.
	L2VPN raw mode with IPV6 inner payload	This scenario is handled using the **lag hash speculate-mpls inner-ipv6-raw** command. Some sections of the IPv6 source and destination address fields are also used for load-balance hashing.

Load balance hash support on different devices

The following table list the support of the Load Balancing Hash configurations on various SLX-OS devices.

Table 22: Support Matrix

Command	Supported In	Not Supported In
• load-balance hash ethernet etype • load-balance hash ethernet vlan	• All platforms	None
• load-balance hash ethernet da-mac • load-balance hash ethernet sa-mac	• Extreme 8720 • Extreme 8520 • SLX 9640 • SLX 9540 • SLX 9250 • SLX 9150	• Extreme 8820 • SLX 9740
• load-balance hash ethernet smac-dmac	• Extreme 8820 • SLX 9740	All other platforms
• load-balance hash ip protocol	• All platforms	None

Table 22: Support Matrix (continued)

Command	Supported In	Not Supported In
• load-balance hash ip dst-ip • load-balance hash ip dst-l4-port • load-balance hash ip src-ip • load-balance hash ip src-l4-port	• Extreme 8720 • Extreme 8520 • SLX 9640 • SLX 9540 • SLX 9250 • SLX 9150	• Extreme 8820 • SLX 9740
• load-balance hash ip sip-dip	• Extreme 8820 • SLX 9740	All other platforms
• load-balance hash ipv6 ipv6-next-hdr	• All platforms	None
• load-balance hash ipv6 ipv6-dst-ip • load-balance hash ipv6 ipv6-dst-l4-port • load-balance hash ipv6 ipv6-src-ip • load-balance hash ipv6 ipv6-src-l4-port	• Extreme 8720 • Extreme 8520 • SLX 9640 • SLX 9540 • SLX 9250 • SLX 9150	• Extreme 8820 • SLX 9740
• load-balance hash ipv6 sipv6-dipv6	• Extreme 8820 • SLX 9740	All other platforms
• load-balance hash l4 src-dst-port	• Extreme 8820 • SLX 9740	All other platforms
• load-balance hash mpls label1 • load-balance hash mpls label2 • load-balance hash mpls label3	• SLX 9540 • SLX 9640	All other platforms
• load-balance hash mpls all-labels	• Extreme 8820 • SLX 9740	All other platforms

LAG hash support on different devices

The following table list the support of the LAG Hash configurations on various SLX-OS devices.

Table 23: Support Matrix

Command	Supported In	Not Supported In
lag hash bos	• SLX 9540 • SLX 9640	All other platforms
lag hash hdr-count	• SLX 9150 • SLX 9250 • SLX 9540 • SLX 9640 • Extreme 8520 • Extreme 8720	• SLX 9740 • Extreme 8820
lag hash hdr-start	• SLX 9150 • SLX 9250 • SLX 9540 • SLX 9640 • Extreme 8520 • Extreme 8720	• SLX 9740 • Extreme 8820
lag hash normalize	• SLX 9540 • SLX 9640 • SLX 9740 • Extreme 8820	All other platforms
lag hash pwctrlword	• SLX 9540 • SLX 9640	All other platforms
lag hash rotate	• SLX 9540 • SLX 9640 • SLX 9740 • Extreme 8820	All other platforms
lag hash speculate-mpls	• SLX 9540 • SLX 9640	All other platforms
lag hash srcport	• SLX 9150 • SLX 9250 • SLX 9540 • SLX 9640 • Extreme 8520 • Extreme 8720	• SLX 9740 • Extreme 8820

MPLS transit load balancing

A hashing scheme for enhanced load balancing allows MPLS transit load balancing to include inner headers of different packet types in parallel. With this enhanced load

balancing scheme, SLX-OS is capable of load balancing the MPLS packets based on the inner headers like Inner source/destination mac address, Inner IPv4 source/destination address, Inner IPv6 source/destination address, Inner L4 port number if the inner header is IPv4.

Note
MPLS transit load balancing is not supported on Extreme 8820, SLX 9740, SLX 9640, and SLX 9540 devices.

This hashing scheme is supported only with the "Layer 2 Optimized" Tcam profile and when the feature is enabled by default in this profile. When the **profile tcam layer2-optimised-1** configuration is activated, the "Error: Operation not supported in the current hardware TCAM profile" message is displayed for the following commands as these functionalities are already taken care of by this enhanced hashing scheme:

- lag hash speculate-mpls inner-eth
- lag hash speculate-mpls inner-ip-raw
- lag hash speculate-mpls inner-ip-tag
- lag hash speculate-mpls inner-ipv6-raw
- lag hash speculate-mpls inner-ipv6-tag

Displaying LAG hashing with "Layer 2 Optimized" Tcam profile

Use the **show port-channel load-balance** command to display the configured parameters for LAG hashing when the **profile tcam layer2-optimised-1** configuration is activated.

```
device# show port-channel load-balance
Header parameters
        Ethernet Mask: sa-mac da-mac etype vlan
        ip: src-ip dst-ip protocol src-l4-port dst-l4-port
        ipv6: ipv6-src-ip ipv6-dst-ip ipv6-next-hdr ipv6-src-l4-port ipv6-dst-l4-port
        mpls: label1 label2 label3

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

mpls_speculate: Enabled

Inner Header parameters for MPLS packets
                Ethernet Mask: Selective 64 bits from sa-mac da-mac vlan
                ip: src-ip dst-ip src-l4-port dst-l4-port
                ipv6: ipv6-src-ip ipv6-dst-ip

load-balance-type hash-based
```

For comparison, the following displays the **show port-channel load-balance** command output for a non layer 2 optimized profile:

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

```
        mpls: label1 label2 label3

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

mpls_speculate:Enabled

load-balance-type hash-based
```

Troubleshooting LAGs

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

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

Troubleshooting static LAGs

If a link is not able to join a static LAG:

- Make sure 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.

Troubleshooting dynamic (LACP) LAGs

If a link is not able to join dynamic (LACP) LAG:

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

Show and clear LAG commands

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

Displaying port-channel information

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

```
device# show port-channel summary
Flags:  D - Down                P - Up in port-channel (members)
        U - Up (port-channel)   * - Primary link in port-channel
        S - Switched
        M - Not in use. Min-links not met
===== =============== ========== ===============
Group Port-channel    Protocol   Member ports
===== =============== ========== ===============
1     Po 1    (D)     None       Eth 2/125 (D)
                                 Eth 4/125 (D)
2     Po 2    (D)     None       Eth 2/126 (D)
                                 Eth 4/126 (D)
10    Po 10   (U)     LACP       Eth 2/4* (P)
                                 Eth 2/18 (P)
100   Po 100  (U)     None       Eth 2/10* (P)
                                          Eth 2/11 (P)
```

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

```
device# show port-channel detail
 LACP Aggregator: Po 10
 Aggregator type: Standard
  Actor System ID - 0x8000,f4-6e-95-9f-13-e2
  Admin Key: 0010 - Oper Key 0010
  Receive link count: 2 - Transmit link count: 2
  Individual: 0 - Ready: 1
  Partner System ID - 0x8000,f4-6e-95-9f-15-a4
  Partner Oper Key 0010
 Flag * indicates: Primary link in port-channel
 Number of Ports: 2
 Minimum links: 1
 Member ports:
   Link: Eth 0/9 (0xC012140) sync: 1
   Link: Eth 0/10 (0xC014140) sync: 1   *
```

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

```
device# show port-channel 10
Port-channel 10 is admin down, line protocol is down (admin down)
Hardware is AGGREGATE, address is 00e0.0c70.cc07
    Current address is 00e0.0c70.cc07
Interface index (ifindex) is 671088650
Minimum number of links to bring Port-channel up is 1
MTU 1548 bytes
LineSpeed Actual     : Nil
Allowed Member Speed : 10000 Mbit
Priority Tag disable
Forward LACP PDU: Enable
Last clearing of show interface counters: 00:29:09
```

```
Queueing strategy: fifo
Receive Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
    Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Time since last interface status change: 00:29:09
```

Displaying LAG hashing

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

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

load-balance-type hash-based
```

Displaying LACP system-id information

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

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

Displaying LACP statistics

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

```
device# show lacp counter
Traffic statistics
Port            LACPDUs            Marker           Pckt err     Sent     Recv
Sent     Recv     Sent     Recv
Aggregator Po 3  Eth 1/6                          110      0            0
0            0        0
```

Clearing LACP counter statistics on a LAG

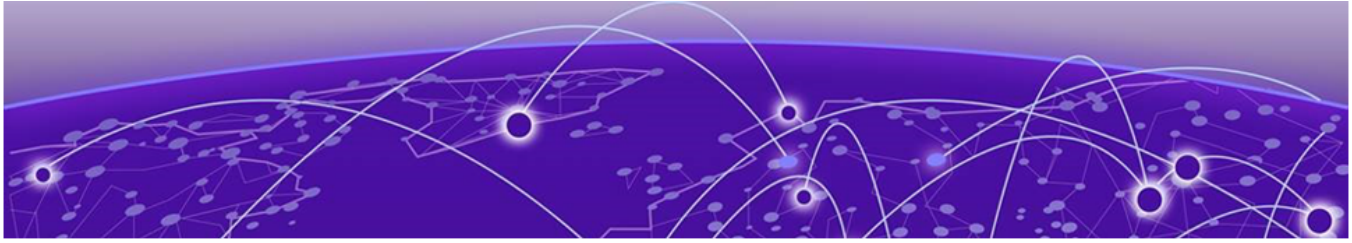
Enter the **clear lacp** *LAG_group_number* **counters** command to clear the LACP counter statistics for the specified LAG group number.

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

Clearing LACP counter statistics on all LAG groups

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

```
device# clear lacp counter
```



VLANs

802.1Q VLAN overview on page 46
Configuring VLANs on page 46
Enabling Layer 3 routing for VLANs on page 52
VLAN statistics on page 53
VE route-only mode on page 55

802.1Q VLAN overview

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

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

Configuring VLANs

The following sections discuss working with VLANs on Extreme devices.

Configuring a VLAN

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

```
device# configure terminal
```

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

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

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

Configuring a switchport interface

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

```
device# configure terminal
```

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

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

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

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

Configuring the switchport interface mode

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

```
device# configure terminal
```

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

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

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

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

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

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

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

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

Configuring the switchport access VLAN type

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

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

```
device# configure terminal
```

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

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

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

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

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

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

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

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

Configuring a VLAN in trunk mode

Ensure that reserved VLANs are not used.

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

```
device# configure terminal
```

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

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

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

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

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

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

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

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

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

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

Configuring a native VLAN on a trunk port

Ensure that reserved VLANs are not used.

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

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

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

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

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

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

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

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

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

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

Enabling VLAN tagging for native traffic

Ensure that reserved VLANs are not used.

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

Table 24: Acceptable frame types and system behavior for native VLANs

	Tagged native VLAN	Untagged native VLAN	No native VLAN
Configuration	switchport trunk tag native-vlan (Default) **and** Globally: **vlan dot1q tag native**	no switchport trunk tag native-vlan **or** Global config: **no vlan dot1q tag native**	switchport mode trunk-no-default-native
Acceptable frame type	VLAN-tagged only	All (tagged and untagged)	VLAN-tagged only

Table 24: Acceptable frame types and system behavior for native VLANs (continued)

Receive untagged	Drop	Forward/flood in native VLAN	Drop
Receive tagged on native VLAN	Forward/flood in native VLAN	Forward/flood in native VLAN	Drop
Transmit on native VLAN	Tagged with native VLAN	Untagged packet	Will not forward on native VLAN

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

```
device# configure terminal
```

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

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

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

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

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

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

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

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

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

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

Displaying the status of a switchport interface

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

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

```
 Ingress filter         : enable
 Acceptable frame types : all
 Default Vlan           : 1
 Active Vlans           : 1
```

Displaying the switchport interface type

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

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

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

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

Verifying a switchport interface running configuration

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

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

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

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

Displaying VLAN information

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

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

2. Enter the **show vlan detail** command to display detailed information.

```
device# show vlan det
VLAN: 1, Name: default
Admin state: ACTIVE, Config status: Static
Number of interfaces: 7
      Eth 0/4, tagged, Static
      Eth 0/3, tagged, Static
      Eth 0/2, tagged, Static
      Eth 0/8, tagged, Static
      Eth 0/6, tagged, Static
      Eth 0/9, untagged, Static
      Po 20, tagged, Static
VLAN: 10, Name: VLAN0010
Admin state: ACTIVE, Config status: Static
Number of interfaces: 3
      Eth 0/3, tagged, Static
      Eth 0/2, tagged, Static
      Eth 0/4, tagged, Static
      Po 20, tagged, Static
VLAN: 11, Name: VLAN0011
Admin state: ACTIVE, Config status: Static
Number of interfaces: 3
      Eth 0/3, tagged, Static
      Eth 0/2, tagged, Static
      Eth 0/4, tagged, Dynamic (MVRP)
VLAN: 12, Name: VLAN0012
Admin state: ACTIVE, Config status: Dynamic (MVRP)
Number of interfaces: 1
      Eth 0/4, tagged, Dynamic (MVRP)
VLAN: 13, Name: VLAN0013
Admin state: ACTIVE, Config status: Dynamic (EP tracking)
Number of interfaces: 1
      Eth 0/6, tagged, Dynamic (EP tracking)
VLAN: 14, Name: VLAN0014
Admin state: INACTIVE(member port down), Config status: Static
Number of interfaces: 1
      Eth 0/8, tagged, Static
```

Enabling Layer 3 routing for VLANs

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a VLAN.

```
device(config)# vlan 200
```

3. Create a virtual Ethernet (VE), assign an IP address and mask, and enable the interface.

```
device(config)# interface ve 200
device(config-Ve-200)# ip address 10.2.2.1/24
device(config-Ve-200)# no shutdown
```

   A VE interface can exist without a VLAN configuration, but it must be provisioned in the VLAN in order to be used.
4. Enter the **router-interface** command and specify the VLAN.

```
device(config-vlan-200)# router-interface ve 200
```

VLAN statistics

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

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

Please note that:

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

Enabling statistics on a VLAN

1. Enter the global configuration mode.

```
device# configure terminal
```

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

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

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

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

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

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

Displaying statistics for VLANs

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

```
device# show statistics vlan

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

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

Table 25: Output descriptions of the show statistics vlan command

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

Displaying VLAN statistics for a specific VLAN

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

```
device# show statistics vlan 10

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

Clearing statistics on VLANs

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

```
device# clear statistics vlan
```

Clearing statistics for a specific VLAN

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

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

VE route-only mode

The MAC learning of dropped packets is not affected by this feature. ARP requests (broadcast), LACP, and BPDU packet processing are also not affected. The following table lists the effects of this mode on a variety of features.

Table 26: Effects of VE route-only mode on features

Feature	Ingress port as route-only port (incoming frames)	Egress port as route-only port (outgoing frames)
Packets requiring switching	Drop, learn MAC address	Forwarded/switched
Packets requiring routing	Forwarded	Forwarded
ARP requests	Trapped/punted, ARP response generated	Forwarded
LACP packets	Trapped/punted and processed	Forwarded
STP/BPDU packets	Trapped/punted and processed	Forwarded

Note the following considerations and limitations:

- Egress packets through a port configured as route-only are transmitted irrespective of whether they are switched or routed.
- This feature is enabled on the active management module (MM), and is available on the other MM after a failover.
- The number of TCAM entries required for this feature depends on the maximum number of physical ports. On a there are approximately 40 physical ports per PPE , and a maximum of 512 LAG ports for the entire system. Therefore the maximum number of TCAM entries for route-only support is 40. These entries are available on all TCAM profiles.

Configuring VE route-only mode on a physical port

1. Enter global configuration mode.
   ```
   device# configure terminal
   ```
2. Create a VLAN.
   ```
   device(config)# vlan 100
   ```
3. Specify an Ethernet interface.
   ```
   device(config)# interface ethernet 0/2
   ```
4. Enter the **switchport** command to configure Layer 2 characteristics.
   ```
   device(conf-if-eth-0/2)# switchport
   ```

5. Specify trunk mode.

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

6. Tag the port to a VLAN.

```
device(conf-if-eth-0/2)# switchport mode trunk allowed vlan add 100
```

7. Enter the **route-only** command to enable Layer 3 routing exclusively on the port.

```
device(conf-if-eth-0/2)# route-only
```

   Use the **no route-only** command to revert to default Layer 2 and Layer 3 behavior.

8. Enable the interface and exit to global configuration mode.

```
device(conf-if-eth-0/2)# no shutdown
device(conf-if-eth-0/2)# exit
```

9. Verify the Ethernet configuration.

```
device(conf-if-eth-0/2)# do show running-cocnfig interface ethernet 0/2
switchport
switchport mode trunk
switchport trunk allowed vlan add 100
route only
no shutdown
```

10. Verify the port statistics for switching packets dropped.

```
device(conf-if-eth-0/2)# do show interface ethernet 0/2
               Ethernet 0/2 is up, line protocol is up (connected)
               Hardware is Ethernet, address is 768e.f80a.033c
               Current address is 768e.f80a.033c
           ….
          Rate info:
               Input 0.001008 Mbits/sec, 0 packets/sec, 0.00% of line-rate
               Output 0.000252 Mbits/sec, 0 packets/sec, 0.00% of line-rate
               Route-Only Packets Dropped: 17
               Time since last interface status change: 21:35:41
```

11. Enter virtual Ethernet (VE) configuration mode and specify the VLAN.

```
device(config)# interface ve 100
```

12. Assign an IP address and mask and enable the interface.

```
device(config-Ve-100)# ip address 10.2.2.2/24
device(config-Ve-100)# no shutdown
```

13. Confirm the VE configuration.

```
device(config-Ve-100)# do show running-config interface ve 100
interface Ve 100
 ip proxy-arp
 ip address 10.2.2.2/24
 no shutdown
```

Configuring VE route-only mode on a LAG port

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a VLAN.

```
device(config)# vlan 100
```

3. Specify a port-channel interface.

```
device# configure terminal
device(config)# interface port-channel 1
```

4. Enter the **switchport** command to configure Layer 2 characteristics.

```
device(config-Port-channel-1)# switchport
```

5. Specify trunk mode.

```
device(config-Port-channel-1)# switchport mode trunk
```

6. Tag the port to a VLAN.

```
device(config-Port-channel-1)# switchport trunk allowed vlan add 100
```

7. Enable tagging on native VLAN traffic.

```
device(config-Port-channel-1)# switchport trunk tag native-vlan
```

8. Enter the **route-only** command top enable Layer 3 routing exclusively on the port.

```
device(config-Port-channel-1)# route-only
```

   Use the **no route-only** command to revert to default Layer 2 and Layer 3 behavior.

9. Enable the interface and exit to global configuration mode.

```
device(config-Port-channel-1)# no shutdown
device(config-Port-channel-1)# exit
```

10. Verify the port-channel configuration.

```
device(config-Port-channel-1)# do show running-cocnfig interface port-channel 1
 interface Port-channel 1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 100,200
 switchport trunk tag native-vlan
 route-only
 no shutdown
```

11. Enter virtual Ethernet (VE) configuration mode and specify the VLAN.

```
device(config)# interface ve 100
```

12. Assign an IP address and mask and enable the interface.

```
device(config-Ve-100)# ip address 10.2.2.2/24
device(config-Ve-100)# no shutdown
```

13. Verify the VE configuration.

```
device(config-Ve-100)# )# do show running interface ve 100
interface Ve 100
 ip proxy-arp
 ip address 10.2.2.2/24
 no shutdown
```

VxLAN Layer 2 Gateway

VxLAN Layer 2 gateway overview on page 58
VxLAN Layer 2 gateway considerations and limitations on page 60
Configuring VxLAN Layer 2 gateway on page 61
VxLAN Layer 2 gateway support for bridge domains on page 63
VxLAN Layer 2 support for LVTEP on page 66

VxLAN Layer 2 gateway overview

A split-horizon topology is supported. As there is no tunnel-to-tunnel flooding of broadcast, unknown unicast, and multicast (BUM) traffic, all nodes participating in the VLAN must be connected through VxLAN tunnels.

The following figure illustrates an example Layer 2 gateway topology.

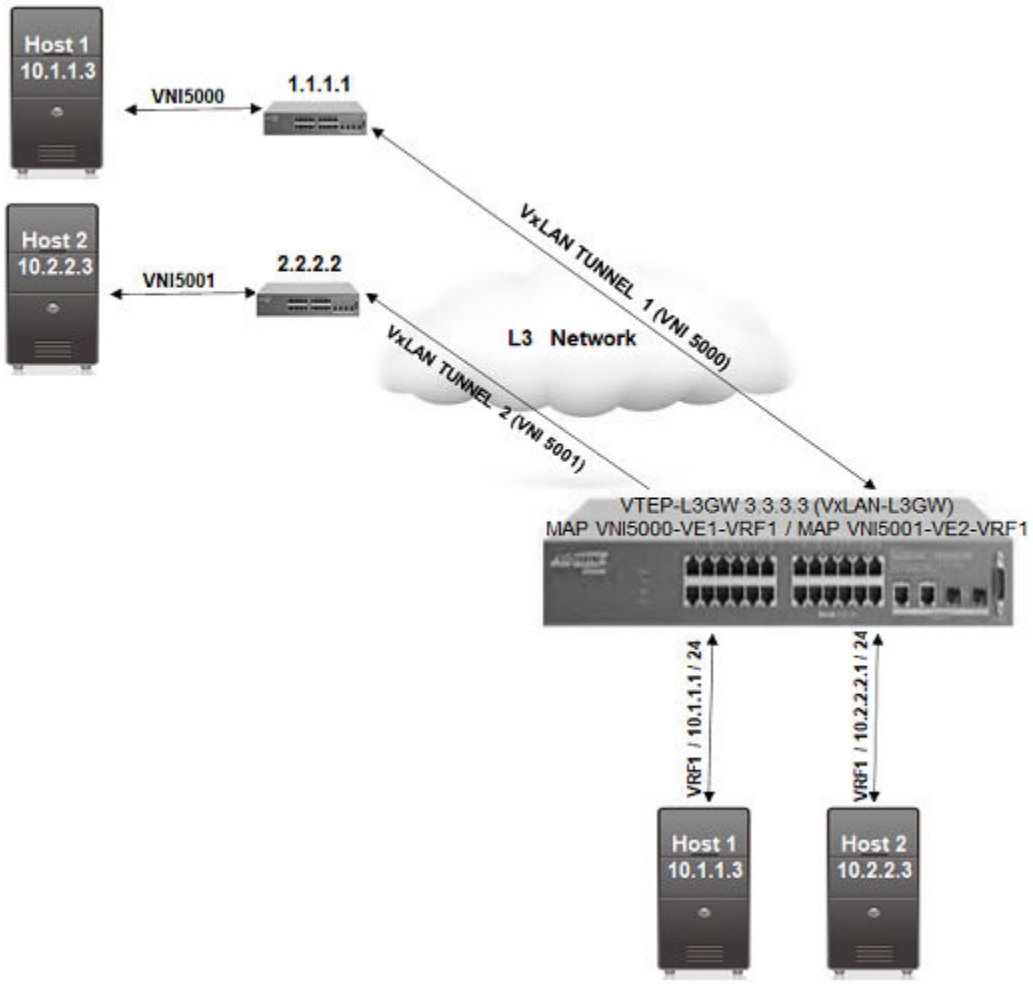


Figure 1: Layer 2 gateway topology

In this topology, device 1, 2, and 3 are all VxLAN Layer 2 gateways. On the device 3, tunnel 1 and tunnel 2 are mapped to VLAN 5, which has two hosts, i.e., MAC 3 and MAC 4 and it is connected to two other hosts of device 1 and device 2, which connect to hosts MAC 1 and MAC 2, respectively, through VxLAN tunnels 1 and 2, respectively. If MAC 3 needs to establish traffic to MAC 1, initially there will be BUM flooding and upon a response from MAC 1, where MAC 1 is learned through tunnel 1. Subsequently the traffic goes directly from MAC 3 to device 1 on tunnel 1. The traffic in the reverse direction comes from device 1 that is decapsulated, and goes to MAC 3.

VLANs on each node are extended through the common Virtual Network Instance (VNI) 5000. The MAC addresses of the local hosts are learned on access points and the MAC addresses are learned on the VxLAN tunnels.

Device	IP address	Mapping
1: SLX 9540 series	1.1.1.1	VNI 5000 < > VLAN 10
2: VDX 6740 series	2.2.2.2	VNI 5000 < > VLAN 20
3: SLX 9540 series	3.3.3.3	VNI 5000 < > VLAN 5

VxLAN Layer 2 gateway considerations and limitations

- For the maximum number of tunnels supported are:
  - 250 for SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720
  - 1024 for SLX 9540/SLX 9640
- It supports maximum of 64 ECMP paths.
- It do not support Layer 2 snooping.
- It also do not support VRRP source IP addresses and Multi-Chassis Trunks (MCTs).
- The QoS DSCP field is not configurable for VxLAN tunnel and its default value is 0. The QoS DSCP for VxLAN at the ingress VTEP is derived from the User Packet. In the Egress VTEP, DSCP for the decapsulated packet is either taken from the Outer header DSCP or inner packet of DSCP value.
- The QoS TTL behaviour follows the Pipe model both at Ingress and Egress VTEPs. The QoS TTL field for VxLAN tunnel is not configurable. The default value is 255, which gets applied to the Outer header for the VxLAN encapsulated packets.
- MTU : MTU field is not configurable. The MTU is based on the IP interface MTU. If a packet is bigger than the IP interface MTU minus the VxLAN header, then the packet is dropped.
- VxLAN tunnels have the standard UDP header encapsulated with the standard defined value of 4789. This value cannot be changed or modified. The SLX-OS expects VxLAN tunnel packets to be received with this value.
- VxLAN tunnels are supported in the default profile.
- VxLAN tunnels are not supported when the counter profile 1 or 4 is configured. These profiles do not allocate hardware resources for TX statistics, which is needed for VxLAN tunnels. (SLX 9540 and SLX 9640 only).
- Static VxLAN Tunnels are not supported on the SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.
- The tunnel TX bytes statistics do not account for the outer VLAN header size.
- When BUM packets received on a tunnel are flooded, split horizon drops the packets to the same tunnel. However, the TX Statistics counter on that tunnel increments.

VNI Mapping

VLAN-VNI mapping is shared by all VxLAN tunnels including the ICL. There are three methods of configuring VNI mapping.

- Auto-Mapping
  - This is the default configuration and is recommended for use with the cluster. The first 32K are reserved for the VNI range. A one-to-one mapping between the VLAN/BD and the VNI is configured.
- Manual Mapping for a specific VLAN/BD (hybrid mode)
  - With **`map vni auto`** configured, a specific VLAN/BD can be manually mapped. The manual mapping VNI range starts at 32768. Other VLANs/BDs will continue to use auto mapping.
- Manual mapping of all VLANs/BDs (disable auto mapping)
  - When auto mapping is disabled, manual VNI mapping is required for all VLANs/BDs created on the system (even the VLANs/BDs that are not configured under EVPN). In this mode, the complete VNI range (1-2^24) is available for manual mapping.

Note

- When the VNI mapping is changed, traffic on the ICL is impacted.
- The VNI Mapping on the two nodes must match.

Configuring VxLAN Layer 2 gateway

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **`overlay-gateway`** command, specify the name of a gateway, and enter VxLAN overlay gateway configuration mode.

```
device(config)# overlay-gateway GW1
```

3. Enter the **`map vlan vni`** command and specify **`l2-extension`**.

```
device(config-overlay-gw-GW1)# map vlan 5 vni 5000
```

4. Enter the **`map bridge-domain`** command and specify a bridge domain and VNI.

```
device(config-overlay-gw-GW1)# map bridge-domain 1 vni 2000
```

5. Enter the **`ip interface`** command and specify a loopback ID.

```
device(config-overlay-gw-GW1)# ip interface loopback 1
```

6. Enter the **`activate`** command to activate the site.

```
device(config-overlay-gw-GW1)# activate
```

7. In global configuration mode, enable EVPN configuration mode and configure the EVPN instance.
   a. Enter default EVPN configuration mode.

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

      Default mode is the only available mode.
   b. Enable the auto-generation of the import and export route-target community attributes for the default EVPN instance.

      ```
      device(config-evpn-default)# route-target both auto
      ```

   c. Enable the auto-generation of a route distinguisher (RD) for the default EVPN instance.

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

   d. Add the BDs to the default EVPN instance.

      ```
      device(config-evpn-default)# bridge-domain add 1-2
      ```

   e. Add the VLANs to the default EVPN instance.

      ```
      device(config-evpn-default)# vlan add 11-12
      ```

8. Configure BGP routing with neighbor and address-family attributes.
   a. In global configuration mode, enable BGP routing and enter BGP router configuration mode.

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

   b. Specify the autonomous system number (ASN) for the AS in which the remote neighbor resides.

      ```
      device(config-bgp-router)# neighbor 7.7.100.7 remote-as 100
      ```

   c. Configure the BGP device to communicate with a neighbor through a specified interface, in this case loopback 1.

      ```
      device(config-bgp-router)# neighbor 7.7.100.7 update-source loopback 1
      ```

   d. Repeat the above two substeps for the other peer address, as in the following example.

      ```
       neighbor 8.8.100.8 remote-as 100
       neighbor 8.8.100.8 update-source loopback 1
      ```

   e. Enable IPv4 and IPv6 unicast address-family.

      ```
      device(config-bgp-router)# address-family ipv4 unicast
      device(config-bgp-router)# address-family ipv6 unicast
      ```

9. Enable the L2VPN address-family configuration mode to configure a variety of BGP EVPN options.
   a. Enable L2VPN address-family configuration mode and enter BGP EVPN configuration mode.

      ```
      device(config-bgp-router)# address-family l2vpn evpn
      ```

   b. Specify VxLAN encapsulation for the first peer.

      ```
      device(config-bgp-evpn)# neighbor 8.8.100.8 encapsulation vxlan
      ```

   c. Enable the exchange of information with BGP neighbors and peer groups.

      ```
      device(config-bgp-evpn)# neighbor 8.8.100.8 activate
      ```

10. In privileged EXEC mode, enter the **show overlay-gateway** command to confirm the gateway configuration.

```
device# show overlay-gateway
Overlay Gateway "GW1", ID 1,
Admin state up
IP address 3.3.3.3 (loopback 1), Vrfdefault-vrf
Number of tunnels 1
Packet count: RX 17909 TX 1247
Byte count : RX (500125) TX 356626
```

11. In privileged EXEC mode, enter the **show tunnel** command to confirm the tunnel configuration.

```
device# show tunnel 61441
Tunnel 61441, mode VXLAN
Ifindex 0x7c00f001, Admin state up, Operstate up
Source IP 3.3.3.3, Vrf: default-vrf
Destination IP 1.1.1.1
Active next hops on node 1:
IP: 4.4.4.5, Vrf: default-vrf
Egress L3 port: Ve45, Outer SMAC: 609c.9f5a.4415
Outer DMAC: 609c.9f5a.0015, ctag: 0
BUM forwarder: yes
```

12. In privileged EXEC mode, enter the **show vlan** command to confirm the VLAN configuration.

```
device# show vlan 5
VLAN             Name            State                       Ports
Classification
(R)-RSPAN                                                    (u)-Untagged
                                                             (t)-Tagged
================ =============== ========================== ===============
====================
5                VLAN05          ACTIVE                      Eth 2/1(t)
                                                             Eth 2/5(t)
                                                             tu61441  vni5000
```

13. In privileged EXEC mode, enter the **show mac-address-table** command to confirm the MAC configuration.

```
device# show mac-address-table
VlanId/BDId    Mac-address       Type      State      Ports/LIF/PW
35 (V)         609c.9f5a.5b15    Dynamic   Active     Po 35
45 (V)         609c.9f5a.4415    Dynamic   Active     Po 45
5 (V)          0000.0400.0011    Dynamic   Active     tu61441
5 (V)          0000.0500.0011    Dynamic   Active     Eth 0/5
5 (V)          0000.0400.0011    Dynamic   Active     tu61441
5 (V)          0000.0500.0011    Dynamic   Active     Eth 0/5
Total MAC addresses : 6
device#
```

VxLAN Layer 2 gateway support for bridge domains

Since a bridge domain supports different port and VLAN endpoints, all of its traffic can be extended to a remote node through one VNI.

Also, VxLAN gateway support to bridge domains enables VLAN translation of traffic on both sides of the network. The local VLANs can use different VLAN tags on either side of the network and map to the same VNI.

Note
Only point-to-multipoint bridge domains can be extended over VxLAN tunnels. Point-to-point bridge domains cannot be extended.

You can extend the bridge domain under a site configuration. You can configure the bridge domain to VNI mapping automatically with auto mode where the bridge domain to the VNI is mapped implicitly. For example, VLANs 1 through 4096 are mapped to VNI 1 through 4090, respectively, and the bridge domain 1 is mapped to 4097. You can also configure the bridge domain to a VNI map manually, similarly to that of a VLAN.

Note
The default tagging mode for a bridge domain is raw mode.

Configuring VxLAN Layer 2 Gateway support for bridge domains

Before performing this configuration, configure a point-to-multipoint bridge domain.

Follow these steps configure a VxLAN Layer 2 gateway to support bridge domains.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a VxLAN overlay gateway and access the overlay gateway configuration mode.

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

3. Specify a loopback interface.

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

4. Enable the mapping of a bridge domain to a VNI.

```
device(config-overlay-gw-gateway1)# map bridge-domain 1 vni 999
```

5. Activate the gateway.

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

The following summarizes the configuration example.

```
device# configure terminal
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)# ip interface loopback 1
device(config-overlay-gw-gateway1)# map bridge-domain 1 vni 999
device(config-overlay-gw-gateway1)# site bd1
device(config-overlay-gateway-gateway1)# activate
```

VxLAN Layer 2 gateway payload tag processing

An SLX-OS device provides the following modes for the processing of the payload tag that is received on the attachment-circuit packets:

- VxLAN RFC-compliant mode
- Enhanced payload tag transport mode

VxLAN RFC-compliant mode

In RFC-compliant mode, the VLAN tag in a packet is not carried in the packet and must be stripped at the ingress device before the VxLAN encapsulated packet is sent into the network.

To configure RFC-compliant mode, configure the bridge domain in raw mode as shown in the following example.

```
pw-profile test
    vc-mode raw

bridge-domain 10 p2mp
   pw-profile test
```

Then extend the bridge domain in the overlay gateway, as shown in the following example.

```
overlay-gateway gateway1
type layer2-extension
ip interface loopback 1
map bridge-domain 10 vni 999
site vcs1
 ip address 10.67.67.1
 extend bridge-domain add 10
```

Note
An SLX-OS device supports RFC-compliant mode with the bridge domain-based VxLAN service only.

Enhanced payload tag transport mode

In enhanced payload tag transport mode, one VLAN tag from the traffic is carried as part of the VxLAN encapsulated packet as an inner payload tag. This tag can carry the PCP value to include the priority information and also can interoperate with other devices. This tag is removed in the remote device capable of this behavior.

Note
This mode does not interoperate with RFC-compliant mode.
This mode is supported for VLAN-based VxLAN service and bridge domain-based VxLAN service with tag mode.

To configure enhanced payload tag transport mode, configure the bridge domain in tagging mode as shown in the following example.

```
pw-profile test
    vc-mode tag
```

```
bridge-domain 10 p2mp
   pw-profile test
```

Then extend the bridge domain in the overlay gateway, as shown in the following example.

```
overlay-gateway gateway1
type layer2-extension
ip interface Loopback 1
map bridge-domain 10 vni 999
site vcs1
 ip address 10.67.67.1
 extend bridge-domain add 10
```

VxLAN Layer 2 support for LVTEP

This section details the support for a logical VxLAN tunnel end point (LVTEP) at Layer 2.

LVTEP control plane

The LVTEP control plane uses MCT Control Plane Designated Forwarder Election among the cluster peers. BGP VxLAN tunnels that are discovered automatically are treated as cluster client end points (CCEPs).

The *VxLAN encapsulation* and *Unicast/Multicast encapsulation over ICL* table shows Ethernet Segment Identifier (ESI) values for the VxLAN tunnels.

The ESI label is allocated globally for the LVTEP and is the same for all LVTEP tunnels. The tunnel operational status that is used for LVTEP tunnels is the same as that used for cluster clients.

Note
The LVTEP is supported with the default TCAM profile.
All the **show** commands that apply to MCT clients are also applicable to the tunnel cluster clients.

LVTEP data plane

Example topology

The following figure illustrates a basic LVTEP topology for the data plane, with cluster nodes supporting remote peers and client end points (CEPs) and cluster client end points (CCEPs).

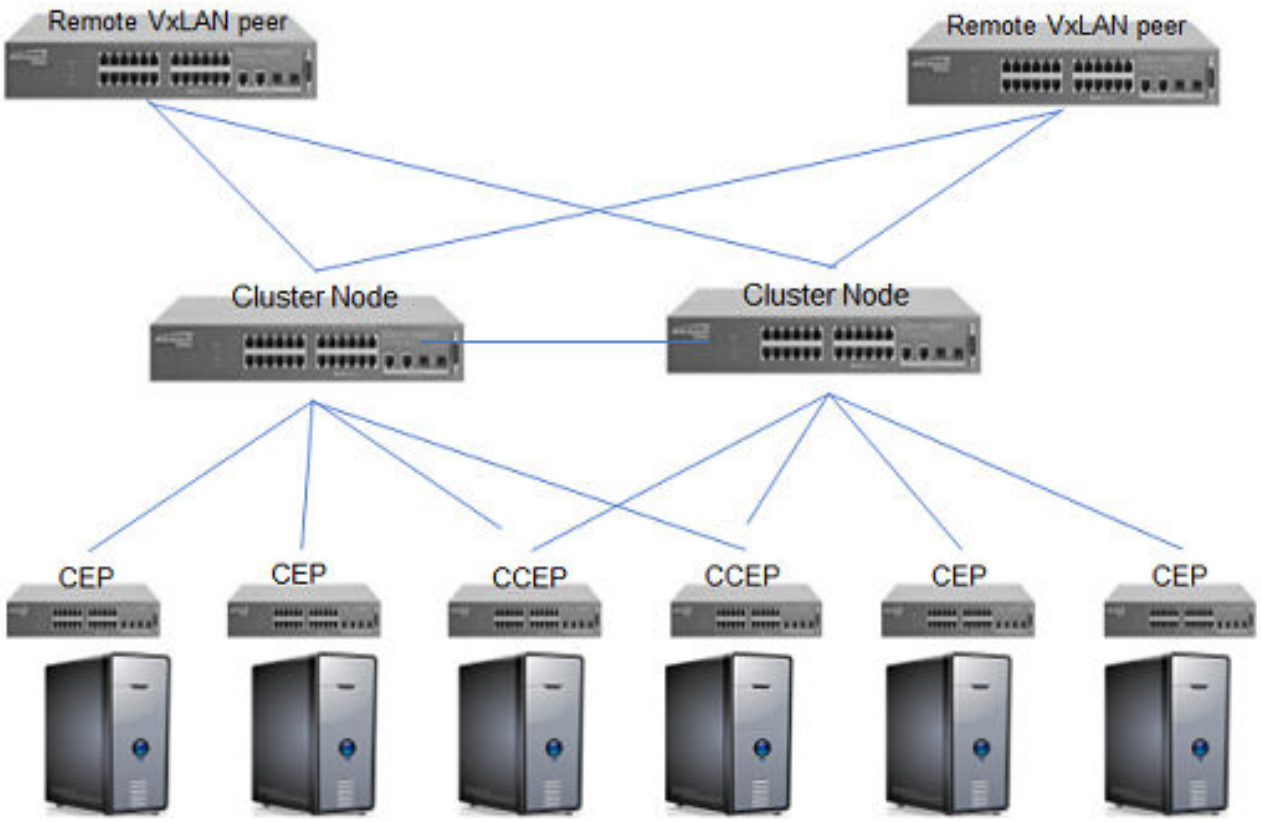


Figure 2: LVTEP topology

Packet formats

The following tables describe the supporting packet formats.

Table 27: VxLAN encapsulation

Outer MAC header	Outer IP header	UDP header	VxLAN header	Original L2 frame

The preceding packet format is used between the remote peers for both unicast and BUM traffic.

Table 28: Unicast/Multicast encapsulation over ICL

Outer MAC header	Outer IP header	UDP header	VxLAN header	Original L2 frame

The preceding packet format is used between the cluster peers for unicast traffic.

The above packet format is used between the cluster peers for BUM traffic when the traffic is received on the CCEP Interface. This applies to both the Layer 2 CCEP interface or the tunnel CCEP interface.

Deployment scenario 1: BGP session and tunnel down, remote peers

For a BGP EVPN deployment, the MAC addresses learned on the VxLAN tunnel are moved to point to the ICL tunnel. Traffic destined to this MAC is sent over the ICL tunnel and forwarded from the MCT peer node.

Port-based VLAN bundle service

The attachment circuit (AC) port is configured with a tag protocol identifier (TPID) that is different from that of the incoming traffic; the port is also configured with an untagged VLAN. All the traffic coming in on this port is treated as untagged traffic. The following figure illustrates this topology.

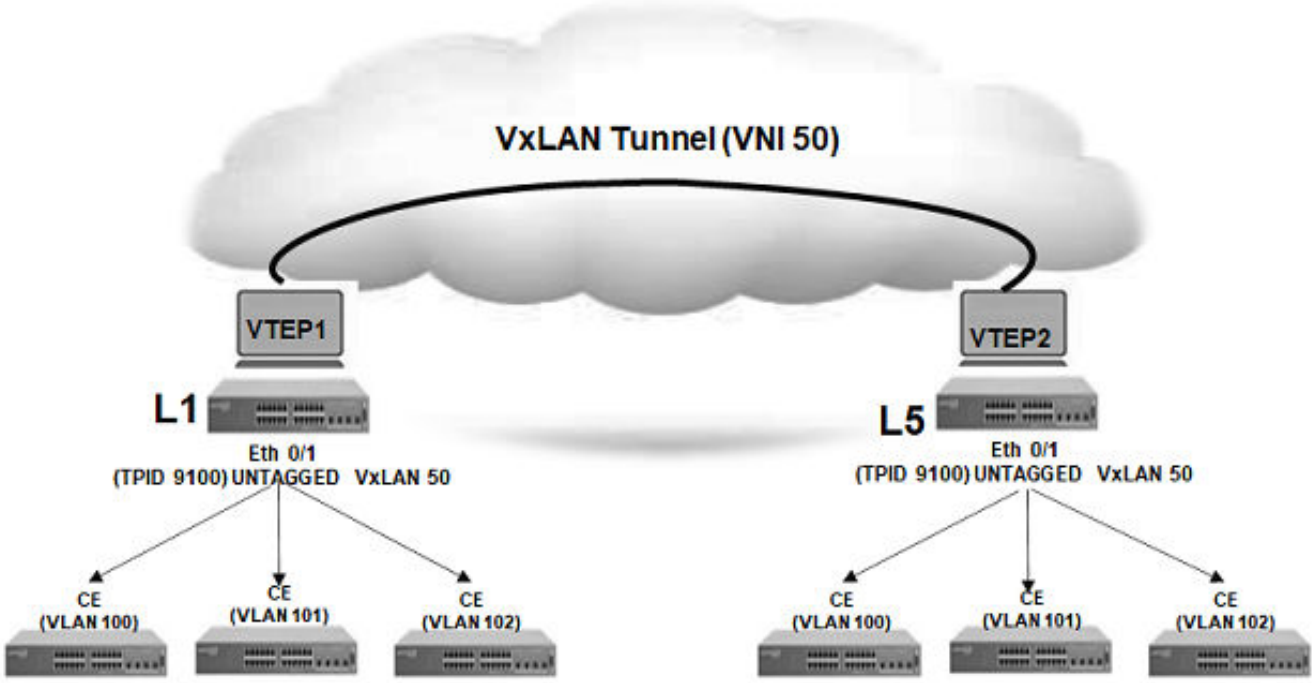


Figure 3: Port-based VLAN bundle service

The untagged VLAN configured on the AC port is extended through the EVPN. All the traffic arriving on this untagged VLAN is bundled with a single Virtual Network Identifier (VNI). The MAC addresses are learned on the untagged VLAN that is configured on that port, irrespective of the VLAN tag of the incoming packet. As all

the traffic on this port is mapped to a single flooding domain, MAC addresses on this VLAN must be unique.

Note
This feature is not supported on the SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.

The following example illustrates a configuration with a VLAN on an Ethernet port.

```
device(conf-if-eth-0/1)# switchport
device(conf-if-eth-0/1)# switchport mode access
device(conf-if-eth-0/1)# switchport access vlan 200
device(conf-if-eth-0/1)# tag-type 9100
```

The following example illustrates a configuration with a bridge domain (BD) logical interface (LIF) on an Ethernet port.

```
device(conf-if-eth-0/1)# switchport
device(conf-if-eth-0/1)# switchport mode trunk-no-default-native
device(conf-if-eth-0/1)# logical-interface ethernet 0/1.1 untagged vlan 200
device(conf-if-eth-0/1)# tag-type 9100
```

Configuring VxLAN LVTEP support

1. Configure multiple loopback interfaces to support BGP neighbor address-family and the LVTEP IP address.
   a. Enter global configuration mode.

      ```
      device# configure terminal
      ```

   b. In global configuration mode, specify a loopback port number.

      ```
      device(config)# interface loopback 1
      ```

   c. Configure a loopback interface with OSPF area 0 and an IP address, and enable the interface to support BGP neighbor address-family.

      ```
      device(config-Loopback-1)# ip ospf area 0
      device(config-Loopback-1)# ip address 6.6.100.6/32
      device(config-Loopback-1)# no shutdown
      ```

   d. Configure a second loopback interface to support the LVTEP IP address.

      ```
      interface Loopback 2
       ip ospf area 0
       ip address 6.7.100.67/32
       no shutdown
      ```

      The same address is used for both nodes in the cluster.
2. In global configuration mode, create two VLANs to support a pair of logical interfaces (LIFs) and BDs.

   ```
   device(config)# vlan 11-12
   ```

3. Configure the LIFs and BDs.
   a. Specify an Ethernet interface.

      ```
      device(config)# interface ethernet 0/5
      ```

   b. Configure the parent interface as switchport.

      ```
      device(conf-if-eth-0/5)# switchport
      ```

c. Specify trunk mode.

```
device(conf-if-eth-0/5)# switchport mode trunk
```

d. Enable the interface.

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

e. Specify a service instance and enter LIF configuration mode.

```
device(conf-if-eth-0/5)# logical-interface ethernet 0/5.1
```

f. Specify an interface and create a dual-tagged (inner VLAN) VLAN.

```
device(conf-if-eth-lif-0/5.1)# vlan 10 inner-vlan 1
```

The VLAN in the LIF configuration is for VLAN tag classification. This example shows a dual-tagged LIF being configured. The expected packet that enters through this port must be dual-tagged, without VLAN 10 and the inner VLAN 1, in order to be classified as a packet received for this LIF.

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

```
device(conf-if-eth-lif-0/5.1)# no shutdown
```

h. Repeat Step 3e through Step 3g for the second logical interface, and specify a second inner VLAN.

```
logical-interface ethernet 0/5.2
  vlan 10 inner-vlan 2
```

4. Create and configure a BD.

a. Create BD 1.

```
device(config)# bridge-domain 1 p2mp
```

By default, the bridge-domain service type is point-to-multipoint (**p2mp**).

b. Bind the logical interfaces for attachment circuit (AC) endpoints to the BD.

```
device(config-bridge-domain-1)# logical-interface ethernet 0/5.1
```

Logical interfaces representing BD endpoints must be created before they can be bound to a BD. For further information, refer to *Logical Interfaces*.

c. Ensure that local switching is enabled for BD 1.

```
device(config-bridge-domain-1)# local-switching
```

Local switching is enabled by default.

d. Enable the dropping of Layer 2 bridge protocol data units (BPDUs) for BD 1.

```
device(config-bridge-domain-1)# bpdu-drop-enable
```

A default pseudowire (PW) profile is automatically configured, with the following defaults:

```
Vc_mode = RAW Mode
mtu = 1500
mtu_enforce = NO
pw_profile_control_word = 0
pw_profile_flow_label = 0
```

e. Repeat the above BD configuration for the second BD, as in the following example.

```
bridge-domain 2 p2mp
 logical-interface ethernet 0/5.2
 pw-profile default
 bpdu-drop-enable
 local-switching
```

5. Configure an overlay gateway.
   a. In global configuration mode, specify a gateway.

```
device(config)# overlay-gateway gw1
```

   b. Specify the type as Layer 2 extension.

```
device(config-overlay-gw-gw1)# type layer-2-extension
```

   c. Specify the LVTEP loopback interface.

```
device(config-overlay-gw-gw1)# ip interface loopback 2
```

   d. Configure the automatic mapping of VLANs/BDs to Virtual Network Identifiers (VNIs).

```
device(config-overlay-gw-gw1)# map vni auto
```

   e. Activate the gateway.

```
device(config-overlay-gw-gw1)# activate
```

6. In global configuration mode, enable EVPN configuration mode and configure the EVPN instance.
   a. Enter default EVPN configuration mode.

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

Default mode is the only available mode.

   b. Enable the auto-generation of the import and export route-target community attributes for the default EVPN instance.

```
device(config-evpn-default)# route-target both auto
```

   c. Enable the auto-generation of a route distinguisher (RD) for the default EVPN instance.

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

   d. Add the BDs to the default EVPN instance.

```
device(config-evpn-default)# bridge-domain add 1-2
```

   e. Add the VLANs to the default EVPN instance.

```
device(config-evpn-default)# vlan add 11-12
```

7. Configure the cluster.
   a. In global configuration mode, specify an MCT cluster name (in this example, "c1") to enable cluster configuration mode.

      ```
      device(config)# cluster c1
      ```

   b. Specify a port channel interface through which to reach the MCT cluster peer.

      ```
      device(config-cluster-c1)# peer-interface port-channel 1
      ```

   c. Specify the IP address of the MCT cluster peer.

      ```
      device(config-cluster-c1)# peer 7.7.100.7
      ```

   d. Exit to Privileged EXEC mode.
8. Configure BGP routing with neighbor and address-family attributes.
   a. In global configuration mode, enable BGP routing and enter BGP router configuration mode.

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

   b. Specify the autonomous system number (ASN) for the AS in which the remote neighbor resides.

      ```
      device(config-bgp-router)# neighbor 7.7.100.7 remote-as 100
      ```

   c. Configure the BGP device to communicate with a neighbor through a specified interface, in this case loopback 1.

      ```
      device(config-bgp-router)# neighbor 7.7.100.7 update-source loopback 1
      ```

   d. Repeat the above two substeps for the other peer address, as in the following example.

      ```
       neighbor 8.8.100.8 remote-as 100
       neighbor 8.8.100.8 update-source loopback 1
      ```

   e. Enable IPv4 and IPv6 unicast address-family.

      ```
      device(config-bgp-router)# address-family ipv4 unicast
      device(config-bgp-router)# address-family ipv6 unicast
      ```

9. Enable the L2VPN address-family configuration mode to configure a variety of BGP EVPN options.
   a. Enable L2VPN address-family configuration mode and enter BGP EVPN configuration mode.

      ```
      device(config-bgp-router)# address-family l2vpn evpn
      ```

   b. Specify VxLAN encapsulation for the first peer.

      ```
      device(config-bgp-evpn)# neighbor 8.8.100.8 encapsulation vxlan
      ```

   c. Enable the exchange of information with BGP neighbors and peer groups.

      ```
      device(config-bgp-evpn)# neighbor 8.8.100.8 activate
      ```

10. Repeat the above steps for the other node in the cluster, with modifications as appropriate.

LVTEP support for other features

AC LIF

AC LIFs of type untagged, single tagged, and double tagged are supported.

Layer 2 ACLs

Layer 2 end points (leaf) support Layer 2 ACLs.

Rate limiting

Layer 2 end points (leaf) do not support rate limiting.

QoS

QoS behavior is similar to that for a single VTEP gateway. Details are listed in the following tables for DiffServe tunneling uniform and pipe modes in an overlay network.

Table 29: QoS behavior for DiffServ tunneling uniform mode

Traffic type	VxLAN origination	VxLAN tunnel termination
VLAN cases (untagged)	DSCP 0 and TTL 255 are sent on the tunnel header.	NA
VLAN cases (tagged)	Priority Code Point (PCP) is mapped to IP DCSP of VxLAN tunnel and TTL is set to 255.	DSCP is remapped to PCP of L2 traffic.
BD (raw, single tagged)	PCP of VLAN is mapped to DSCP and TTL is set to 255.	DCSP is remapped to PCP of VLAN.
BD (raw, double tagged)	PCP of outer VLAN is mapped to DSCP and TTL is set to 255.	DCSP is remapped to PCP of both inner and outer VLAN. Original PCP inner VLAN is not retained.
BD (raw, untagged)	DSCP 0 and TTL 255 are sent on the tunnel header	NA
BD (tagged, single tagged)	PCP is mapped to DSCP and TTL is set to 255	DSCP is remapped to PCP of L2 traffic.
BD (tagged, double tagged)	Outer VLAN PCP is mapped to DSCP and TTL is 255 Inner VLAN header is carried with original PCP.	DSCP is remapped to PCP of outer VLAN and Inner VLAN of L2 traffic.
BD (tagged, untagged)	DSCP 0 and TTL 255 are sent on the tunnel header. Dummy VLAN is added: VLAN ID is BD ID and PCP is 0.	NA

Table 30: QoS behavior for DiffServ tunneling pipe mode

Traffic type	VxLAN origination	VxLAN tunnel termination
VLAN cases (untagged)	DSCP 0 and TTL 255 are sent on the tunnel header.	NA
VLAN cases (tagged)	Priority Code Point (PCP) is mapped to IP DCSP of VxLAN tunnel and TTL is set to 255.	DSCP is remapped to PCP of L2 traffic.

Table 30: QoS behavior for DiffServ tunneling pipe mode (continued)

Traffic type	VxLAN origination	VxLAN tunnel termination
BD (raw, single tagged)	PCP of VLAN is mapped to DSCP and TTL is set to 255.	DCSP is remapped to PCP of VLAN.
BD (raw, double tagged)	PCP of outer VLAN is mapped to DSCP and TTL is set to 255.	DCSP is remapped to PCP of both inner and outer VLAN. Original PCP inner VLAN is not retained.
BD (raw, untagged)	DSCP 0 and TTL 255 are sent on the tunnel header	NA
BD (tagged, single tagged)	PCP is mapped to DSCP and TTL is set to 255	DSCP is remapped to PCP of L2 traffic.
BD (tagged, double tagged)	Outer VLAN PCP is mapped to DSCP and TTL is 255 Inner VLAN header is carried with original PCP.	DSCP is remapped to PCP of outer VLAN and Inner VLAN of L2 traffic.
BD (tagged, untagged)	DSCP 0 and TTL 255 are sent on the tunnel header. Dummy VLAN is added: VLAN ID is BD ID and PCP is 0.	NA

Note
In QoS behavior for DiffServ tunneling pipe mode table - If the packet has IP Header, post-decap packet IP DSCP value is taken from the inner DSCP value of the incoming packet in the VxLAN tunnel termination traffic.

MTU

MTU behavior is similar to that for a single VTEP gateway.

Inner packet tag behavior

Inner packet tag behavior is similar to that for a single VTEP gateway.

The following table summarizes this behavior for VLANs and BDs.

Table 31: Inner packet tag behavior for VLANs and BDs

VLAN/BD Configuration	Inner packet tag behavior	Remarks
VLAN	Inner packet is always untagged.	
BD raw mode	Inner packet is always untagged.	This is RFC-compliant behavior.
BD tagged mode	Inner packet is always tagged.	This mode can be used if a tag must always be sent in the inner packet.

Statistics

The LVTEP tunnel CCEP supports statistics. For BUM traffic, although the traffic is suppressed (through split horizon), it is still accounted for as part of the statistics. This behavior is the same as existing single VTEP behavior.

Nondefault TPID

The TPID field is located at the same position as the EtherType/length field in untagged frames, and is thus used to distinguish the frame from untagged frames. If you require support for dual tagging or provider backbone bridge (PBB), the outer TPID of the packet must be configured to a value different from the default.

The raw pass-through support for an untagged LIF requires you to configure the interface TPID to a value that allows it to treat all traffic received on that port as untagged. You can configure the TPID on port and LAG interfaces.

Important
When the tag type is changed on interface, the interface is brought down first, causing all learned MAC addresses to be flushed.

Hardware limitations of the SLX-9540/SLX-9640

The TPID feature has the following limitations:

- **Tagging**: The TPID configuration is supported for an outer tag. If dual-tagging needs to be supported on the interface, the inner tag must be 0x8100.
- **TPID**: Up to four TPIDS are supported system-wide. One is used by default (TPID = 0x8100) and cannot be changed.
- **LSP FRR**: Hardware support for LSP FRR is available only for TPID 0x8100. If you require a label switched path with fast reroute (LSP FRR) configuration, note that none of the routable interfaces (whether a router port or a LIF of a VE) can have any nondefault TPID configuration, because FRR always assumes that the link layer has the default TPID of 0x8100.

Note
The LSP FRR limitation is for any tag-type configured in the device. You can configure either FRR or tag-type on any interface in the device, as in the following example.

```
device(config-router-mpls-lsp-to-avalanche-1)# frr
                        %Error: Not allowed, when a non-default TPID (tag-type)
is configured on any port-channel or physical interfaces.
                        device(config-router-mpls-lsp-to-avalanche-1)#
```

Hardware Limitation of SLX 9740/Extreme 8820

The TPID feature has the following limitation:

- **TPID**: Up to two TPIDS are supported system-wide. One is used by default (TPID = 0x8100) and cannot be changed. Therefore, only one addition TPID is available for user configuration. For dual tagged packets, the inner TPID must always be the default TPID (0x8100).

Configuring a nondefault TPID

Perform the following steps to configure a nondefault TPID. The interface can be a port or a port-channel (LAG).

1. Do the following to configure a nondefault TPID on an Ethernet interface.
   a. Enter global configuration mode.

```
device# configuration terminal
```

   b. Enter interface configuration mode and specify an Ethernet interface.

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

   c. Use the **tag-type** command to configure the TPID.

```
device(conf-if-eth-0/1)# tag-type 0x9100
```

   By default, all interfaces in the system have default TPID value of 0x8100.
   d. Enter the **show interface ethernet** command to confirm the configuration.

```
device# show interface ethernet 0/1
Ethernet 0/1 is up, line protocol is up (connected)
Hardware is Ethernet, address is 609c.9f5f.5005
    Current address is 609c.9f5f.5005
Pluggable media present
Interface index (ifindex) is 203431936
MTU 1548 bytes
10G Interface
LineSpeed Actual     : 1000 Mbit
LineSpeed Configured : Auto, Duplex: Full
Tag-type: 0x9100
Priority Tag disable
Forward LACP PDU: Disable
Route Only: Disabled
Last clearing of show interface counters: 23:12:47
Queueing strategy: fifo
Receive Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    0 packets, 0 bytes
    Unicasts: 0, Multicasts: 0, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
    Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Route-Only Packets Dropped: 0
Time since last interface status change: 23:12:45
```

   e. To revert to the default TPID value, enter the **no tag-type** command.

```
device(conf-if-eth-0/1)# no tag-type
```

   You can achieve the same result by configuring a tag-type of 0x8100.

2. Do the following to configure a nondefault TPID on a port-channel interface.
   a. Enter global configuration mode.

```
device# configuration terminal
```

   b. Enter interface configuration mode and specify a port-channel interface.

```
device(config)# interface port-channel 20
```

   c. Use the **tag-type** command to configure the TPID.

```
device(config-Port-channel-20)# tag-type 0x88a8
```

   By default, all interfaces in the system have default TPID value of 0x8100.
   d. Enter the **show interface port-channel** command to confirm the configuration.

```
device# show interface port-channel 20
Port-channel 20 is up, line protocol is up
Hardware is AGGREGATE, address is 609c.9f5c.ac07
    Current address is 609c.9f5c.ac07
Interface index (ifindex) is 671088660 (0x28000014)
Minimum number of links to bring Port-channel up is 1
MTU 1548 bytes
LineSpeed Actual     : 300000 Mbit
Allowed Member Speed : 100000 Mbit
Priority Tag disable
Forward LACP PDU: Disable
Route Only: Disabled
Tag-type: 0x88a8
Last clearing of show interface counters: 2d01h50m
Queueing strategy: fifo
Receive Statistics:
    34579 packets, 4201368 bytes
    Unicasts: 0, Multicasts: 34579, Broadcasts: 0
    64-byte pkts: 0, Over 64-byte pkts: 17288, Over 127-byte pkts: 17291
    Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
    Over 1518-byte pkts(Jumbo): 0
    Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
    Errors: 0, Discards: 0
Transmit Statistics:
    34578 packets, 4201240 bytes
    Unicasts: 0, Multicasts: 34578, Broadcasts: 0
    Underruns: 0
    Errors: 0, Discards: 0
Rate info:
    Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
    Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Route-Only Packets Dropped: 0
Time since last interface status change: 1d23h59m
```

Configuring TCAM profiles to support LVTEP

1. Enter global configuration mode and enter the **hardware** command.

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

2. In hardware configuration mode, enter the **profile tcam** command.

```
device(config-hardware)# profile tcam default
```

3. Save the running configuration to the startup configuration.

```
device# copy running-config startup config
```

4. Reboot the device.

LVTEP show commands

Cluster show commands

show cluster

```
device# show cluster

Cluster default
===============
Cluster State: Active/Shutdown
Bringup Delay: 10 seconds
Configured Member Vlan Range: All
Active Member Vlan Range: 1-352,456-765,783,785-795
Configured Member BD Range: All
Active Member BD Range: 3-5,26-42
No. of Clients: 4

 Peer Info:
 ----------
 Peer IP: 10.1.1.26, State: Up
 Peer Interface: Ethernet 0/41, Source IP: 10.1.1.28

 Keep-Alive:
 ==========
 IP: 24.1.1.26, State: Down (Source-interface Down)
 Interface: Ethernet 0/20 (default-vrf), Source IP: 22.1.1.28
 Interval: 100ms
 Role: Secondary

 Client Info:
 ------------
 Interface         Id             Description      Local/Remote State       Exceptions
 ---------         --             ------------        -------------------      -----------
 Ethernet 0/11     11             Spirent 3/3         Up   / Down              Remote Down
 Port-channel 20   1020           Server_U27          Up   / Up
 Tunnel 32771      30.30.30.30                        Up   / Up                VLAN
mismatch
 PW                34816                              Down / Down              Client
Down
```

show cluster client

```
device# show cluster client 22
Client Info:
------------
Interface: Port-channel 20, client-id: 1020, Deployed, State: Up
Description          : U27
Local Vlans          : 1,61-300
Local Bridge Domains    : 20,100
Remote Vlans         : 1,61-300
Remote Bridge Domains      : 20
DF Elected for all vlans/BDs
```

BGP show commands

show bgp evpn routes type inclusive-multicast detail

```
device# show bgp evpn routes type inclusive-multicast detail
Total number of BGP EVPN IMR Routes : 12
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
Route Distinguisher: 6.6.100.6:32779
1       Prefix: IMR:[0][IPv4:6.6.100.6],  Status: BL,  Age: 1h54m39s
         NEXT_HOP: 0.0.0.0, Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 0
         AS_PATH:
            Extended Community: RT 100:1073741835
            Adj_RIB_out count: 2,  Admin distance 0
            L2 Label: 11
            RD: 6.6.100.6:32779

Route Distinguisher: 7.7.100.7:32779
5       Prefix: IMR:[0][IPv4:7.7.100.7],  Status: BI,  Age: 1h53m37s
         NEXT_HOP: 7.7.100.7, Metric: 1, Learned from Peer: 7.7.100.7 (100)
          LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 0
         AS_PATH:
            Extended Community: RT 100:1073741835 RT 100:268435456 RT 100:0
ExtCom:03:0c:00:00:00:00:0a:00
            PMSI Attribute Flags: 0x00000000  Label-Stack: 817163 Tunnel-Type: 6 Tunnel-
IP: 0.0.0.0
            Extended Community: ExtCom: Tunnel Encapsulation (Type MPLS)
            L2 Label: 817163
            RD: 7.7.100.7:32779

Route Distinguisher: 8.8.100.8:32779
9       Prefix: IMR:[0][IPv4:8.8.100.8],  Status: BI,  Age: 1h53m29s
         NEXT_HOP: 8.8.100.8, Learned from Peer: 8.8.100.8 (100)
          LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 0
         AS_PATH:
            Extended Community: RT 100:1073741835 RT 100:268435467 RT 100:11
ExtCom:03:0c:00:00:00:00:08:00
            PMSI Attribute Flags: 0x00000000  Label-Stack: 11 Tunnel-Type: 6 Tunnel-IP:
8.8.100.8
            Extended Community: ExtCom: Tunnel Encapsulation (Type Vxlan)
            L2 Label: 11
            RD: 8.8.100.8:32779
```

show bgp evpn routes type mac detail

```
device# show bgp evpn routes type mac detail
Total number of BGP EVPN MAC Routes : 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
Route Distinguisher: 6.6.100.6:36865
1       Prefix: MAC:[0][0000.0607.0000],  Status: BL,  Age: 0h0m53s
         NEXT_HOP: 0.0.0.0, Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 0
         AS_PATH:
            Extended Community: RT 100:1073745921
            Adj_RIB_out count: 2,  Admin distance 0
            L2 Label: 4097
            ESI : 00.000000000000000000
            RD: 6.6.100.6:36865
Route Distinguisher: 8.8.100.8:32779
2       Prefix: MAC:[0][0000.0807.0000],  Status: BI,  Age: 0h5m16s
```

```
         NEXT_HOP: 8.8.100.8, Learned from Peer: 8.8.100.8 (100)
          LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 0
         AS_PATH:
            Extended Community: RT 100:1073741835 RT 100:268435467 RT 100:11
ExtCom:03:0c:00:00:00:00:08:00
            Extended Community: ExtCom: Tunnel Encapsulation (Type Vxlan)
            L2 Label: 11
            ESI : 00.000000000000000000
            RD: 8.8.100.8:32779
```

Tunnel show commands

show tunnel brief

```
device# show tunnel brief
Tunnel 32771, mode VXLAN, unicast
Admin state up, Oper state up
Source IP 6.7.100.67 (  Loopback 2 ), Vrf default-vrf
Destination IP 8.8.100.8
```

show tunnel

```
device# show tunnel 32771
Tunnel 32771, mode VXLAN, unicast
Ifindex 0x7c00f001, Admin state up, Oper state up
Overlay gateway "gw1", ID 1
Source IP 6.7.100.67 (  Loopback 2 ), Vrf default-vrf
Destination IP 8.8.100.8
Configuration source BGP-EVPN
MAC learning BGP-EVPN
Tunnel QOS mode UNIFORM
Active next hops on node 1:
    IP: 10.6.8.8, Vrf: default-vrf
    Egress L3 port: Ve 3, Outer SMAC: 609c.9f5a.3d15
    Outer DMAC: 609c.9f5a.4515, ctag: 0
    BUM forwarder: yes

Packet count: RX 167993610       TX 995395
Byte count  : RX 29902862580     TX 226950060
```

QoS for VxLAN Layer 2 gateways

A VxLAN L2 gateway can interconnect tenants in the same subnet through the VxLAN configuration. The *VxLAN L2 gateway interconnection of tenants in the same subnet* figure as shown below illustrates a simple use case where a packet is sent from VM1. If it is a BUM packets, Leaf-A floods through the packets to all the VTEPs in the same VxLAN segment, until the MAC table at Leaf-A is populated with corresponding entries through mechanisms such as EVPN. Meanwhile, the known unicast packets is forwarded in unicast mode to the corresponding VTEP only.

At Ingress-VTEP, incoming packet's VLAN PCP derives the Traffic-Class (TC). Derived TC is then used to derive the outer DSCP value for VxLAN Encapsulated packet which is sent out via the Tunnel Interface.

At the Egress-VTEP, incoming VxLAN encapsulated packet's outer DSCP value is used to derive the Traffic-Class(TC). Post decapsulation the packets VLAN PCP is derived from the TC derived above and the packets DSCP value is derived based on the **qos-dscp-mode** configuration.

For information on how to use the **qos-dscp-mode** command, refer the *Extreme SLX-OS Command Reference* for this release.

For VxLAN tunnel, the `qos-dscp-mode` configuration has no effect at ingress-VTEP.

MCT ICL tunnels are Internal VxLAN tunnels. The behaviour explained about TC selection, DSCP and VLAN-PCP calculation for overlay-gateway is applicable to MCT Internal Tunnel as well. MCT ICL Tunnel QoS model is `Pipe mode` always. As these are Internal VxLAN Tunnels, `Uniform Mode` is not applicable.

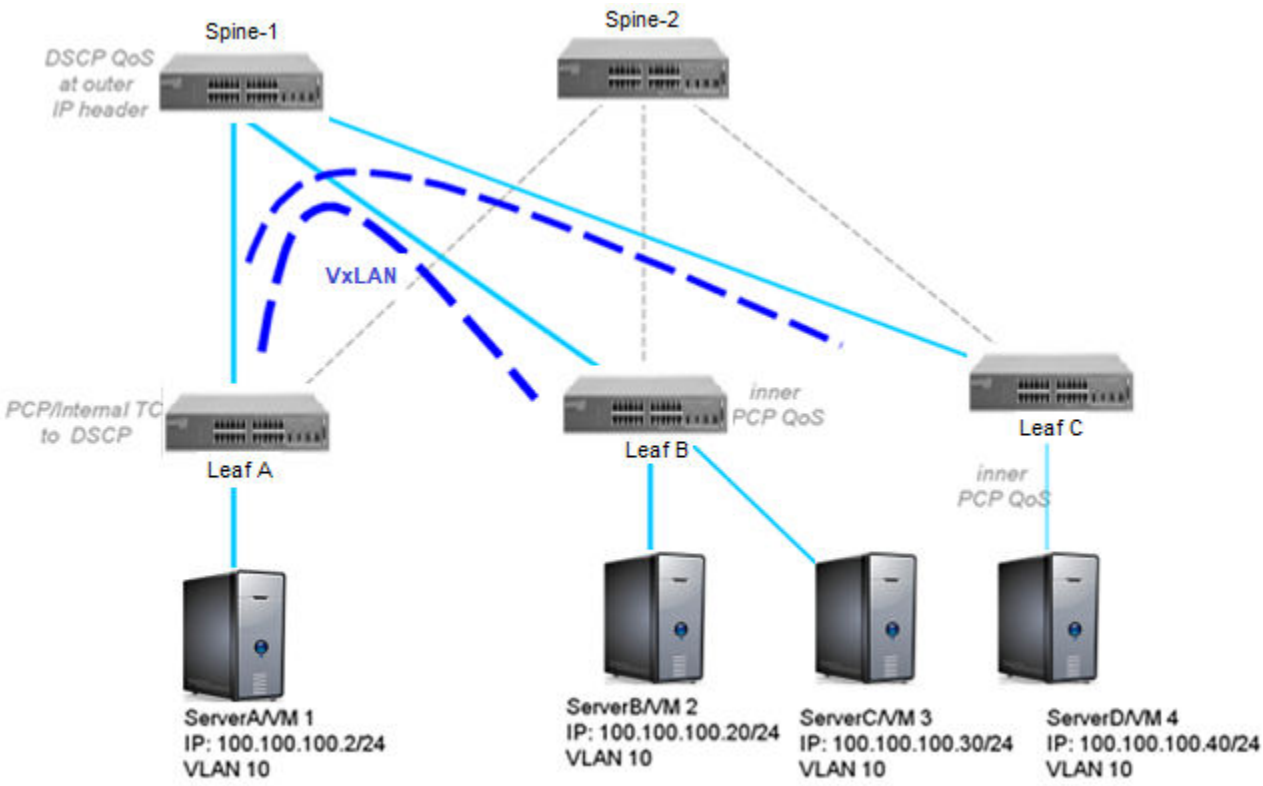


Figure 4: VxLAN L2 gateway interconnection of tenants in the same subnet

VxLAN QoS TTL:

- VxLAN being virtualization Tunnel, VxLAN TTL model is “Pipe Mode” always. “Pipe mode” function is set both at Ingress-VTEP and Egress -VTEP.
- AT Ingress-VTEP, VxLAN Header Outer TTL is a constant value (255). Inside the encapsulated packet, the incoming packet TTL (Payload TTL) is retained same.
- Egress -VTEP validates VxLAN Header Outer TTL for non-Zero value, then discards it. Post decapsulation, the Payload TTL value is retained same without any changes.
- The MCT ICL Tunnels are Internal VxLAN Tunnel. VxLAN TTL behavior explained for Overlay-GW is applicable to MCT ICL internal Tunnel. MCT ICL internal Tunnel TTL model is “Pipe Mode” always.

Configuring QoS for VxLAN Layer 2 gateways

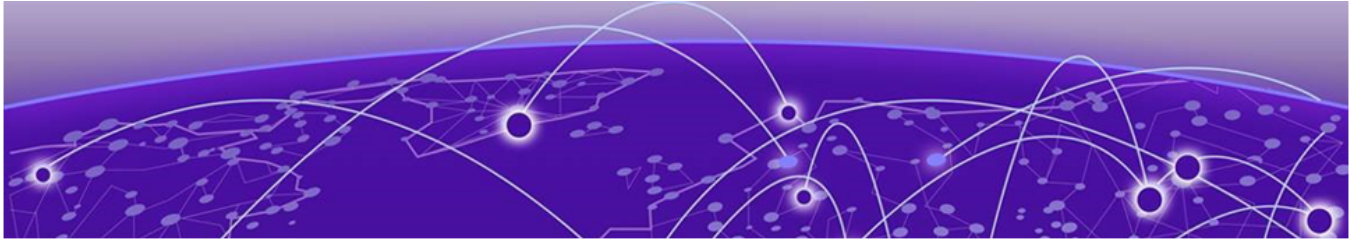
Enter the **`qos-dscp-mode pipe`** command to set the QoS type mode to pipe, which is the default, when configuring the VxLAN L2 gateway. For example, enter the following commands when configuring the gateway for the ingress (tunneling) side of packet forwarding:

```
SLX# conf t
Entering configuration mode terminal
SLX(config)# overlay-gateway  1
SLX(config-overlay-gw-1)# qos-dscp-mode ?
Possible completions:
  pipe      Pipe
  uniform   Uniform
```

```
SLX(config-overlay-gw-1)# qos-dscp-mode pipe ?
Possible completions:
  <cr>
SLX(config-overlay-gw-1)# qos-dscp-mode pipe
```

Note

For information about QoS configuration on VxLAN L3 gateways and on VLAN L2 and L3 gateway interconnections, refer to "QoS for VxLAN Layer 3 gateways" and "QoS for VxLAN Layer 2 and Layer 3 gateway interconnections".



Multiple VLAN Registration Protocol (MVRP)

Multiple VLAN Registration Protocol overview on page 84
Enabling MVRP on an Ethernet interface on page 86
Configuring an edge port on page 87
Configuring the VLAN registration mode on the interface on page 88
MVRP over a port channel on page 88
Configuring the MVRP join, leave, and leave-all timers on page 90
Displaying MVRP configuration information, statistics, and attributes on page 91
Clearing MVRP statistics on page 93

Multiple VLAN Registration Protocol overview

An MVRP-aware device can exchange VLAN configuration information with other MVRP-aware devices, prune unnecessary broadcast and unknown unicast traffic, and dynamically create and manage VLANs on the devices. These devices form a reachability tree that is a subset of an active topology. To avoid any information loop, the forwarding ports of base spanning tree form the active topology.

Note
It is not mandatory to enable any of the STP variants before configuring MVRP if the topology is already Layer 2 loop free where MVRP can still function effectively.

MVRP on the device propagates the configured VLANs to all MVRP participants to declare it over their respective MVRP-enabled ports. The device initiates a join event information which is placed inside an MVRP data unit (MVRPDU). The MVRPDU is transmitted as untagged Ethernet frame and is sent out as an advertisement through all ports wherever a declaration is made. The advertisement is transmitted only when the port's spanning tree state is forwarding.

Similarly, if a VLAN is removed from a port, MVRP on the device propagates a leave event to all MVRP participants which removes the registration and declaration for the VLAN and then sends out a leave message over their respective MVRP enabled ports. The receiving node mimics this behavior and, through this method, the removal of the VLAN configuration is communicated to the entire topology.

VLAN registration occurs only on ports of intermediate nodes where the MVRP advertisement is received. Each registration acts as a pointer towards the source of the VLAN declaration. If the VLAN configuration on a device changes, MVRP automatically changes the VLAN configurations of the affected devices. Using VLAN pruning, MVRP avoids unnecessary flooding and provides solution for better resource utilization and bandwidth conservation.

MVRP considerations and limitations

- You can enable MVRP only on interfaces configured as switchport in both trunk and access modes.
- The interface must be configured in 'switchport access mode' only on the edge ports of the SLX-OS device where MVRP selects statically configured VLANs for propagation. The interfaces connecting the devices on the SLX devices should be only configured in 'switchport trunk mode' where MVRP dynamincally configures the VLANs. MVRPDUs received on ports configured in access mode are dropped; VLANs are neither learned nor propagated.
- MVRP can be learned for all allowed configurable VLANs, up to 4090 VLANs in number.
- MVRP can be enabled on all physical and PO interfaces without any CLI restrictions.
- A maximum of 2k dynamic VLANs are supported on each MVRP-enabled interface. However, the MVRP timer values require adjustment to less aggressive values based on the number of MVRP-enabled interfaces, VLANs to be learned dynamically on each associated interface, load on the system, and the topology.

  If MVRP is scaled to an extent with the default timer settings that are too aggressive, the device may become unresponsive and severe performance impact could occur. Extreme recommends that you increase the MVRP timer values to values that are less aggressive for effective MVRP operation.
- If you manually configure a VLAN that is currently a dynamically-learned VLAN, the VLAN is converted to the Static type and the device initiates an appropriate Syslog message. Though the VLAN is Static in type, its member switch ports are of the Dynamic type (dynamically learned).

  If you configure a Static VLAN on a MVRP-enabled dynamically-learned switch port, the ports are configured to the Static type and the device initiates an appropriate Syslog message.

  You cannot delete a Static VLAN with Dynamic ports through the CLI until all Dynamic ports are converted to Static.
- The device also supports the following Layer 2 features with MVRP:
  - Spanning Tree Protocol (STP)
  - Rapid Spanning Tree Protocol (RSTP)
  - Common Internal Spanning Tree (CIST)
- The following features are not supported with MVRP:
  - PVST and PVST+
  - RPVST and RPVST+

- MSTIs for MSTP
- MVRP over any ring protocols
- MVRP with topology groups

Enabling MVRP on an Ethernet interface

Perform the following steps to enable MVRP on an Ethernet interface.

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

```
device# configure terminal
```

2. Add a VLAN on a device that will be added to the interface.

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

This command accesses VLAN configuration mode.

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

3. Access global configuration mode.

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

4. Enable MVRP globally on the device.

```
device(config)# protocol mvrp
```

This command accesses MVRP configuration mode.

```
device(config-mvrp)#
```

5. Access global configuration mode.

```
device(config-mvrp)# exit
```

6. Configure an Ethernet interface and access interface configuration mode.

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

7. Place the interface in Layer 2 mode.

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

8. Set the Layer 2 interface as trunk.

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

Trunk mode makes the port linkable to other switches and routers.

9. Add the VLAN to the interface as tagged.

```
device(conf-if-eth-0/1)# switchport trunk allowed vlan add 10
```

10. Enable MVRP on the interface.

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

11. Enable the interface.

```
device(conf-if-eth-0/1)# no shutdown
```

12. Display the MVRP status of the configuration.

```
device(conf-if-eth-0/1)# do show mvrp
-------------------------------------------------------------------------------------------
------
Total configured mvrp ports     :       1
Global Status                   :       Enabled
Join-timer(in centiseconds)     :       20
Leave-timer(in centiseconds)    :       100
Leaveall-timer(in centiseconds) :       1000
```

```
-------------------------------------------------------------------------------------------
------
MVRP Port(s): ethe 0/1
```

The following example provides the steps in the previous configuration.

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# exit
device(config)# protocol mvrp
device(config-mvrp)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# switchport
device(conf-if-eth-0/1)# switchport mode trunk
device(conf-if-eth-0/1)# switchport trunk allowed vlan add 10
device(conf-if-eth-0/1)# mvrp enable
device(conf-if-eth-0/1)# no shutdown
device(conf-if-eth-0/1)# do show mvrp
-------------------------------------------------------------------------------------------
---
Total configured mvrp ports     :       1
Global Status                   :       Enabled
Join-timer(in centiseconds)     :       20
Leave-timer(in centiseconds)    :       100
Leaveall-timer(in centiseconds) :       1000
-------------------------------------------------------------------------------------------
---
MVRP Port(s): ethe 0/1
```

Configuring an edge port

This procedure assumes that MVRP is enabled on the device.

Perform the following steps.

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

   ```
   device# configure terminal
   ```

2. Configure an Ethernet interface and access interface configuration mode.

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

3. Enable MVRP on the interface.

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

4. Configure the applicant mode of the interface to non-participant.

   ```
   device(conf-if-eth-0/1)# mvrp applicant-mode non-participant
   ```

   The default applicant mode setting for this command is normal-participant.

   Note
   To ensure that the MVRPDU is not exchanged on the edge port, set the applicant state as non-participant on the port.

The following example are the steps in the previous configuration.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# mvrp enable
device(conf-if-eth-0/1)# mvrp applicant-mode non-participant
```

Configuring the VLAN registration mode on the interface

This procedure assumes that MVRP is enabled on the device.

Note

You cannot configure a static VLAN as forbidden. If a VLAN is configured in the forbidden list and then it is configured as a static VLAN on the device, this VLAN is implicitly removed from the forbidden list.

Perform the following steps.

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

```
device# configure terminal
```

2. Configure an Ethernet interface and access interface configuration mode.

```
device(config)# interface ethernet 0/9
```

3. Enable MVRP on the interface.

```
device(conf-if-eth-0/9)# mvrp enable
```

4. Register the VLAN on the forbidden list.

```
device(conf-if-eth-0/9)# mvrp registration-mode forbidden vlan add 10
```

   You can also provide a range of VLANs.

The following example are the steps in the previous configuration.

```
device# configure terminal
device(config)# interface ethernet 0/9
device(conf-if-eth-0/9)# mvrp enable
device(conf-if-eth-0/9)# mvrp registration-mode forbidden vlan add 10
```

MVRP over a port channel

You can enable MVRP on a port channel configured in either static or dynamic mode. When MVRP is enabled, it runs on the port-channel interface which determines its MAP context but not on individual member ports of the port-channel interface. MVRP derives its port state from the STP state of the port-channel interface for its advertising decisions.

Enabling MVRP over a port channel

Perform the following steps to enable MVRP over a port channel.

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

```
device# configure terminal
```

2. Enable MVRP globally on the device.

```
device(config)# protocol mvrp
```

   This command accesses MVRP configuration mode.

```
device(config-mvrp)#
```

3. Access global configuration mode.

```
device(config-mvrp)# exit
```

4. Configure an Ethernet interface.

```
device(config)# interface ethernet 0/1,2
```

   This command accesses interface configuration mode.

```
device(conf-if-eth-0/1,2)#
```

5. Enable static link aggregation on the interface.

```
device(conf-if-eth-0/1,2)# channel-group 10 mode on
```

6. Enable the interface.

```
device(conf-if-eth-0/1,2)# no shutdown
```

7. Access global configuration mode.

```
device(conf-if-eth-0/1,2)# exit
```

8. Configure the port-channel interface and access port-channel configuration mode.

```
device(config)# interface Port-channel 10
```

9. Enable the port channel in Layer 2 mode.

```
device(config-Port-channel-10)# switchport
```

10. Set trunk mode on the port channel.

```
device(config-Port-channel-10)# switchport mode trunk
```

11. Enable MVRP on the port channel.

```
device(config-Port-channel-10)# mvrp enable
```

12. Enable the port channel.

```
device(config-Port-channel-10)# no shutdown
```

13. Display the MVRP status of the configuration.

```
device(config-Port-channel-10)# do show mvrp
-------------------------------------------------------------------------------------------
------
Total configured mvrp ports     :       1
Global Status                   :       Enabled
Join-timer(in centiseconds)     :       20
Leave-timer(in centiseconds)    :       100
Leaveall-timer(in centiseconds) :       1000
-------------------------------------------------------------------------------------------
------
MVRP Port(s): Po10
```

The following example provides the steps in the previous configuration.

```
device# configure terminal
device(config)# protocol mvrp
device(config-mvrp)# exit
device(config)# interface ethernet 0/1,2
device(conf-if-eth-0/1,2)# channel-group 10 mode on
device(conf-if-eth-0/1,2)# no shutdown
device(conf-if-eth-0/1,2)# exit
device(config)# interface Port-channel 10
device(config-Port-channel-10)# switchport
device(config-Port-channel-10)# switchport mode trunk
device(config-Port-channel-10)# mvrp enable
device(config-Port-channel-10)# no shutdown
device(config-Port-channel-10)# do show mvrp
-------------------------------------------------------------------------------------------
---
Total configured mvrp ports     :       1
```

```
Global Status                    :       Enabled
Join-timer(in centiseconds)      :       20
Leave-timer(in centiseconds)     :       100
Leaveall-timer(in centiseconds) :       1000
-------------------------------------------------------------------------------------------
---
MVRP Port(s): Po10
```

Configuring the MVRP join, leave, and leave-all timers

By default, the MVRP join, leave, and leave-all timer have the following settings.

- The join timer default setting is 20 centiseconds (cs).
- The leave timer default setting is 100 cs. The leave timer setting must be greater than or equal to twice the join timer setting plus 30 centiseconds.
- The leave-all timer default setting is 1000 cs. The leave-all timer setting must be a minimum of three times the value of the leave timer setting.

You can configure global timer settings or settings for each interface that overrides the global timer settings. If the network radius is large or the expected system load is higher normally, Extreme recommends that you change the timer values to higher numbers as appropriate for the MVRP operation to reduce the MVRP PDU exchanges and processing by the NSM.

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

   ```
   device# configure terminal
   ```

2. To set the global timer settings, access MVRP configuration mode.

   ```
   device(config)# protocol mvrp
   ```

3. Set the global MVRP join, leave and leave-all timers values that will be applied implicitly on all the MVRP-enabled interfaces.

   ```
   device(config-mvrp)# timer join 40 leave 200 leave-all 2000
   ```

   In this step, the join timer is set to 40 centiseconds (cs), the leave timer is set to 200 cs, and the leave-all timer is set to 2000 cs.

4. Access global configuration mode.

   ```
   device(config-mvrp)# exit
   ```

5. To set the timers for an individual interface, access the configuration mode for the interface.

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

   This step is for an Ethernet interface. You can also change the settings for a port-channel interface.

6. Configure the timers for the interface.

   ```
   device(conf-if-eth-0/1)# timer join 50 leave 300 leave-all 3000
   device(conf-if-eth-0/1)#
   ```

The following example provides the steps for setting the MVRP timers globally and for an interface.

```
device# configure terminal
device(config)# protocol mvrp
device(config-mvrp)# timer join 40 leave 200 leave-all 2000
device(config-mvrp)# exit
```

```
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# timer join 50 leave 300 leave-all 3000
device(conf-if-eth-0/1)#
```

Displaying MVRP configuration information, statistics, and attributes

You can display the following MVRP information:

- Global and specified interface configuration information
- Interface statistics
- Attributes for all interfaces, a specified interface, or a specified VLAN

Displaying the global MVRP information on the device

To display the global MVRP information on the device including configured ports, global enable status, and timer settings, use the **show mvrp** command.

```
device# show mvrp
-------------------------------------------------------------------
Total configured mvrp ports            : 5
Global Status                          : Enabled
Join-timer(in centiseconds)            : 20
Leave-timer(in centiseconds)           : 100
Leaveall-timer(in centiseconds)        : 1000
-------------------------------------------------------------------
MVRP Port(s): eth0/1, eth0/5, eth0/7, eth0/9, po11
```

Displaying the MVRP information for an Ethernet or port-channel interface

To display the MVRP information for an Ethernet or port-channel interface, use the **show mvrp inteface** command. This information includes the MVRP status, timer and applicant-mode settings, and information about registered, declared, and forbidden VLANs.

```
device# show mvrp interface ethernet 0/1
-----------------------------------------------------------------------------------------
MVRP Status                            : Enabled
Join-timer(in centiseconds)            : 20
Leave-timer(in centiseconds)           : 100
Leaveall-timer(in centiseconds)        : 1000
P2p                                    : Yes
Applicant Mode                         : normal-participant
-----------------------------------------------------------------------------------------
Registered Vlan(s)                     : 1 to 60 77 100 to 500 999
Declared Vlan(s)                       : 1 to 60 77 100 to 500 999
Forbidden Vlan(s)                      : 10
-----------------------------------------------------------------------------------------
```

Displaying the MVRP statistics for an Ethernet or port-channel interface

To display the statistics for received and transmitted MVRPDU messages on the interfaces, use the **show mvrp statistics** command.

The following example displays the statistics for all interfaces.

```
device# show mvrp statistics
Port : eth0/1
-------------------------------------------------------------------------------
Message type                    Transmitted  Received
-------------------------------------------------------------------------------
New                             0            0
In                              0            1809
Join In                         1816         0
Join Empty                      1788         0
Empty                           0            771
Leave                           99           0
Leave-all                       264          512
-------------------------------------------------------------------------------
Total PDUs                      1827         1293
-------------------------------------------------------------------------------
Port : Po100
-------------------------------------------------------------
Message type                    Transmitted  Received
-------------------------------------------------------------------------------
New                             0            2
In                              693          0
Join In                         1800         1777
Join Empty                      0            1956
Empty                           396          0
Leave                           0            96
Leave-all                       369          346
-------------------------------------------------------------------------------
Total PDUs                      1807         1799
-------------------------------------------------------------------------------
```

Displaying MVRP attributes

To display attributes for all or specific MVRP-enabled Ethernet or port-channel interfaces including the port and VLAN states, use the **show mvrp attributes** command.

The following example displays MVRP attributes for all interfaces.

```
device# show mvrp attributes
Port : eth0/17   State : Disabled
-------------------------------------------------------------------------------
VLAN       Registrar            Registrar            Applicant
           State                Mgmt                 State
-------------------------------------------------------------------------------
Port : eth0/5    State : Forwarding
-------------------------------------------------------------------------------
VLAN       Registrar            Registrar            Applicant
           State                Mgmt                 State
-------------------------------------------------------------------------------
1          In                   Fixed                Quiet Active
5          Empty                Normal               Quiet Active
10         In                   Fixed                Quiet Active
Port : po10      State : Forwarding
-------------------------------------------------------------------------------
VLAN       Registrar            Registrar            Applicant
           State                Mgmt                 State
-------------------------------------------------------------------------------
1          In                   Fixed                Quiet Active
5          In                   Normal               Very Anxious Observer
10         Empty                Normal               Quiet Active
```

The following example displays MVRP attributes for a specified interface.

```
device# show mvrp attributes interface ethernet 0/5

Port : eth0/5    State : Forwarding
-------------------------------------------------------------------------------
VLAN       Registrar            Registrar            Applicant
           State                Mgmt                 State
-------------------------------------------------------------------------------
1          In                   Fixed                Quiet Active
5          Empty                Normal               Quiet Active
10         In                   Fixed                Quiet Active
```

The following example displays MVRP attributes for a specified VLAN.

```
device# show mvrp attributes vlan 10
-------------------------------------------------------------------------------
PORT   VLAN      Registrar         Registrar        Applicant
                 State             Mgmt             State
-------------------------------------------------------------------------------
eth0/5 10        In                Fixed            Quiet Active
po10   10        Empty             Normal           Quiet Active
```

Clearing MVRP statistics

To clear MVRP statistics displayed by the **show mvrp statistics** command, use the **clear mvrp statistics** command. You can use this command for either an interface or all interfaces.

- The following example clears statistics for all the MVRP-enabled interfaces.
  ```
  device# clear mvrp statistics
  ```
- The following example clears the MVRP statistics for an interface.
  ```
  device# clear mvrp statistics interface ethernet 0/1
  ```

Multi-Chassis Trunking (MCT)

MCT Overview on page 94
MCT configuration considerations on page 106
Configuring MCT on page 107
Configuring additional MCT cluster parameters on page 109
Displaying MCT information on page 110
VPLS and VLL MCT on the SLX 9640 and 9540 devices on page 112
Layer 3 routing over MCT on page 124
Using MCT with VRRP and VRRP-E on page 128
MCT Use Cases on page 131

MCT Overview

Note
The SLX-OS device does not support Layer 2 protocols over MCT. Spanning Tree Protocol (STP) is disabled by default and must not be enabled with MCT. You must provide a loop-free topology.

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

With MCT, member links of the LAG trunk are connected to two MCT-aware devices. A configuration between the devices enable data flow and control messages between them to establish a logical Inter-Chassis Link (ICL). In this model, if one MCT device fails, a data path remains through the other device.

SLX-OS Layer 2 MCT cluster control protocol (CCP) synchronizes MAC, ARP, IGMP, and cluster management data between the MCT peers, for node resiliency and faster convergence.

On the SLX devices, the data plane is established using a VxLAN tunnel between MCT peers.

SLX-OS MCT provides Layer 3 protocol support for ARP, IPv4 and IPv6 BGP, OSPF, PIM, IGMP, ND6, and IS-IS through a VLAN or bridge domain VE interface. IPv4 or IPv6 Virtual Routing Redundancy Protocol (VRRP) and VRRP Extended (VRRP-E). is also supported.

SLX-OS MCT is only supported in the Default or Apptelemetry Hardware TCAM Profile.

MCT terminology

MCT peer devices	A pair of SLX-OS device configured as peers. The LAG interface is spread across two MCT peer devices and acts as the single logical endpoint to the MCT client. **Note:** MCT is supported across the same chassis type only; for example, SLX-9540 <---> SLX-9540.
MCT Cluster Client	The MCT Cluster Client is the device that connects with the MCT peer devices. It can be a switch or an endpoint server host in the single-level MCT topology, or another pair of MCT switches in a multi-tier MCT topology.
MCT Cluster Control Protocol	The control plane for Layer 2 MCT on the SLX-OS device.
MCT Cluster Client Edge Port (CCEP)	Ports connected to the dual-homed clients. CCEP Ports could be a LAG or a physical port.
MCT Cluster Edge Port (CEP)	Orphan or edge ports connected to one of the MCT nodes.
Inter-Chassis Link (ICL)	Inter-Chassis Link which connects two MCT nodes. For SLX-OS devices, the ICL is a VxLAN tunnel created between the MCT peer that forms the data path.
MCT VLANs	VLANs that are shared by the MCT peers. These VLANs are explicitly configured in the MCT configuration.
Designated Forwarder (DF)	MCT node that is elected to send BUM traffic to a tunnel client for particular VLAN or BD.

SLX-OS MCT control plane

The control plane session requires a direct link btween the nodes (peer-interface) to be operational.The control plane session is used to exchange the following information:

- MCT client discovery
- MCT client VLAN/BD configuration information
- Route exchanges – MAC, MAC/IP, IGMP
- Keep-alive peer IP in auto discovery mode
- Other required state/signaling for protocols, such as PIM, and mac-move detection

A cluster control protocol (CCP) session is used to exchange MAC, MAC/IP and IGMP routes between the MCT peers. A cache is created within the MCT process to hold MAC, ARP and IGMP routes. For the VLANs/BDs that are shared between the MCT nodes, all local routes received from L2, ARP and multicast modules are stored in L2RIB and advertised to the MCT peer. Routes in L2RIB have the following possible sources:

- Locally learned
- MCT Peer

A route selection algorithm is run to determine the best route and download only that route for installation in hardware. L2RIB also generates a sequence number for the routes and passes this on to BGP for MAC mobility and MAC dampening purposes. L2RIB does not include a dampening logic and relies on the MAC move detection feature to detect and mitigate layer-2 loops.

Local routes from l2/arp/multicast modules are cached and advertised to mct-peer and BGP based on vlan/bd extension. Similarly remote routes from MCT peer and BGP are cached in mct process and advertised to L2/ARP/Multicast processes for hardware programming.

Inter-Chassis Link for the SLX Devices

The underlay interface carrying the traffic can be any physical port or port-channel Layer 3 interface between the MCT peers. VE's are not supported. By default, all MCT VLANs or bridge domains (BDs) are extended to the MCT peer.

By default, VLAN-VNI mapping is automatically configured for the ICL VxLAN tunnel. Since a single VLAN-VNI mapping domain is supported, any change to this mapping under the overlay gateway changes the mapping for the ICL and temporarily affects its traffic.

Designated Forwarder Election

The Designated Forwarder (DF) for a given VLAN/BD is the MCT node responsible for forwarding BUM traffic received in the flooding domain. This method is used for VxLAN tunnel clients and PW clients. The DF election is made per client and per VLAN/BD.

When the client state is up on both MCT peers, one MCT node becomes DF for all odd-numbered VLANs/BDs and the other becomes DF for all even-numbered VLANs/BDs. For a PW client, this is run globally, whereas for VxLAN clients this is run per VxLAN tunnel.

If the VxLAN tunnel client is down on one side, the corresponding tunnel on the peer node becomes the DF for all VLANs/BDs extended on it.

MCT Data Plane for the SLX Devices

For the discussion of the SLX MCT data plane, refer to the following topology diagram.

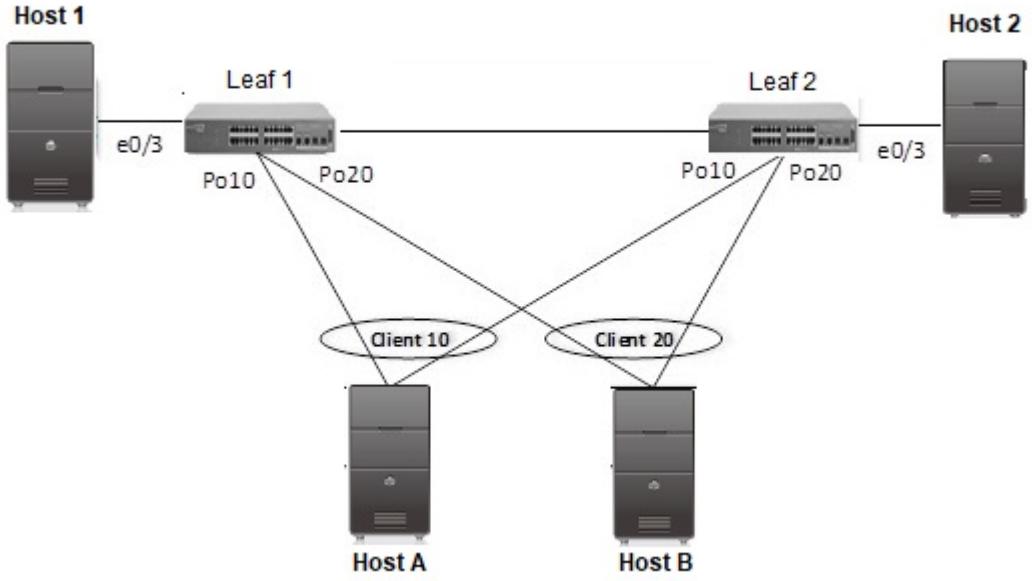


Figure 5: MCT data plane topology example

- Leaf 1 and Leaf 2 are MCT peers.
- Host A and Host B are client nodes configured with Client 10 and 20, respectively. They are connected to Leaf1 and Leaf2 through cluster client edge ports (CCEPs) Po10 and Po20.
- Host 1 and Host 2 are connected to edge ports and are not dual homed. They are connected to Leaf1 and Leaf2 through cluster edge port (CEP) Ethernet 0/3.

Forwarding unicast traffic

For the unicast traffic from the CCEP to the CEP, refer to the following figure.

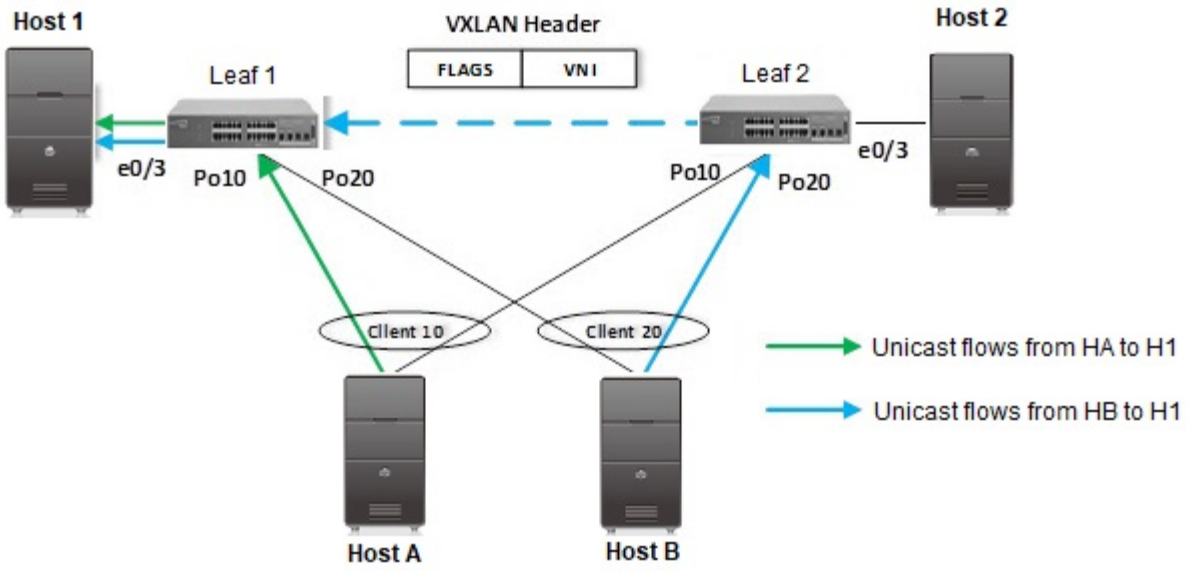


Figure 6: Unicast forwarding traffic from the CCEP to the CEP

- Unicast traffic from Host A and Host B to Host 1 can be hashed to Leaf 1 and Leaf 2.
- Leaf 1 learns the MAC address of Host 1. This traffic is locally switched.
- Leaf 2 has the remote MAC address of Host 1 against the tunnel. This traffic is sent over the ICL.
- The MAC addresses of Host A and Host B are learned as CCL and CCR pointing to the local client port (Po10 and Po20).

For the unicast traffic from the CEP to the CCEP, refer to the following figure.

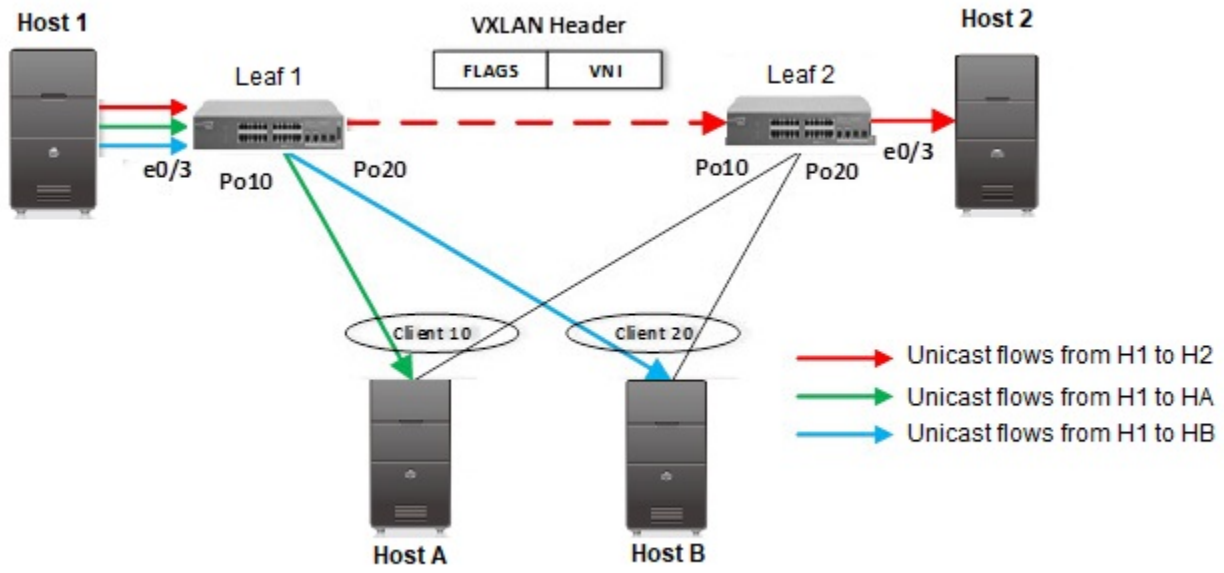


Figure 7: Unicast forwarding traffic from the CEP to the CCEP

- Leaf2 learns the MAC address of Host 2 that is synchronized to Leaf 1. The Unicast traffic from Host 1 to Host 2 is sent over the ICL tunnel to Leaf2 and then is locally switched.
- Based on the learned CCL and CCR MAC addresses, Leaf1 locally switches the unicast traffic from Host 1 to Host A and Host B.

If the local client port goes down, the client MAC address is reprogrammed in the hardware to point to the ICL tunnel. Then the traffic is sent over the ICL to Leaf2 where the learned CCL and CCR MAC addresses ensure the traffic is switched to the client.

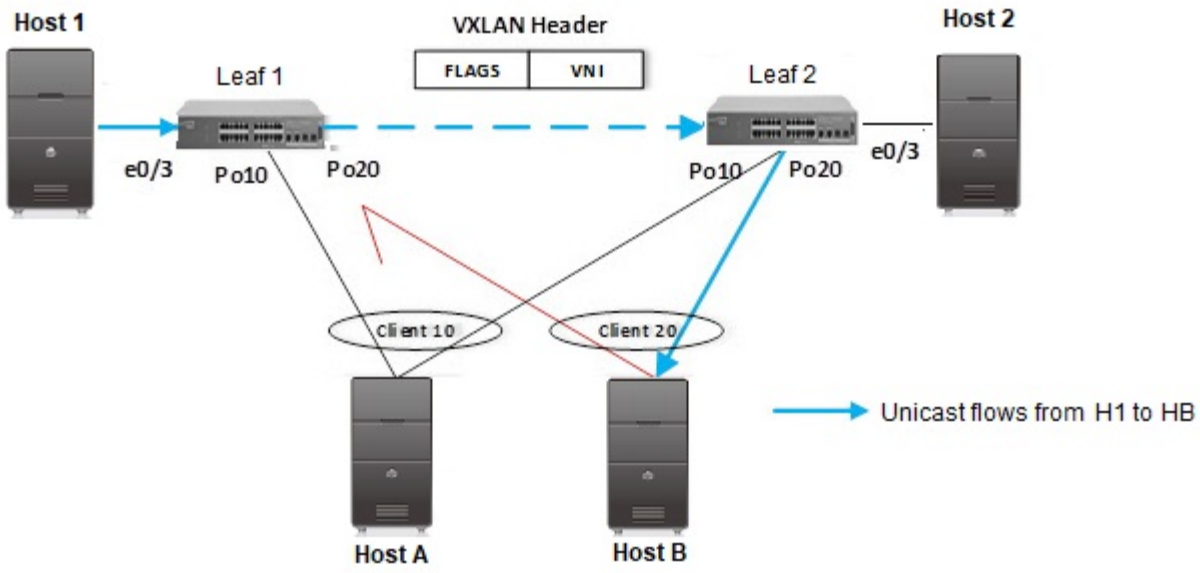


Figure 8: Unicast forwarding traffic from the CEP to the CCEP (local client down)

Flooding traffic from a CEP

For BUM traffic received from Host 1 (CEP) on Leaf 1:

- Leaf1 sends a copy to Leaf2 over the VxLAN tunnel. This copy is encapsulated with a VxLAN header with a VNI.
- Leaf 2 decapsulates the packet and floods the packet to Host 2.
- Copies to Host A and Host B are suppressed on Po10 and Po20.
-

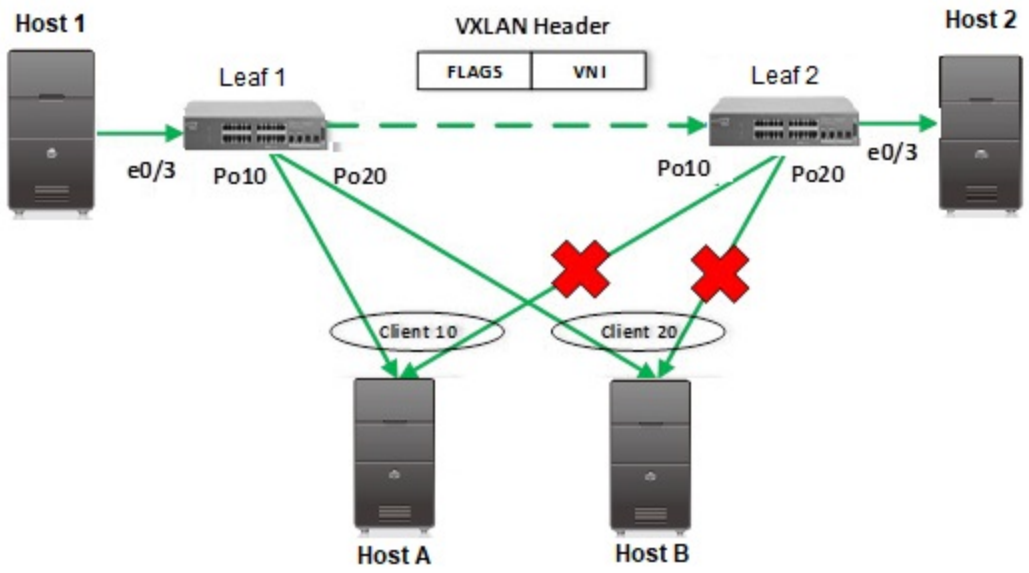


Figure 9: BUM forwarding between cluster edge ports

Flooding traffic received from the CCEP

Traffic received from CCEP Host A on Leaf 1 is flooded to Host 1 over Ethernet 0/3 (Layer 2 flooding), and to Leaf 2 over the VxLAN tunnel with Client 10. Traffic is forwarded to Host B over Po20.

Then Leaf 2 floods the packet to Host 2 over Ethernet 0/3, and but traffic to Host A and Host B over Po10 and Po20 is supressed.

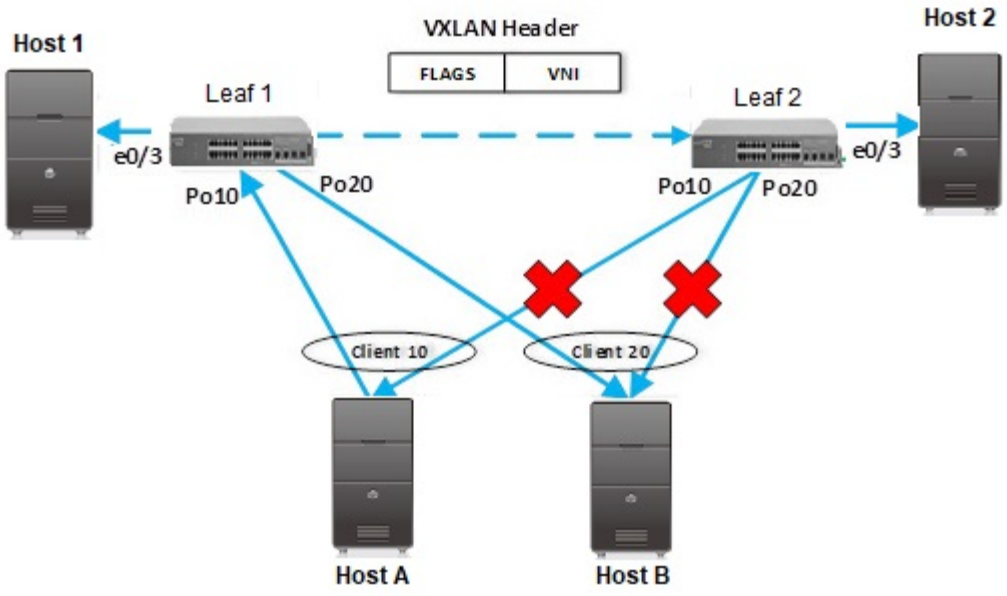


Figure 10: Flooding traffic from the CCEP with DF selection

MAC management

In MCT topologies, MAC addresses can be learned either on local CCEP or CEP interfaces, or from the remote MCT node. If the same MAC is learned from both MCT nodes, then MAC entries learned locally have higher priority than the one learned from the remote peer.

For remote MAC addresses, aging is disabled, and they can only be deleted when delete notifications are received from the remote node that advertised it before for learning.

The following terminologies are associated with MCT MAC entries.

- Dynamic—MAC addresses learned locally on CEP ports
- Cluster Remote (CR)—MAC addresses learned on remote CEP ports
- Cluster Client Local (CCL)—MAC addresses learned locally on a client interface
- Cluster Client Remote (CCR)—MAC addresses learned on a remote client interface

Static MAC handling

Configuration of static MAC entries is allowed over MCT enabled VLANs and CEP and CCEP interfaces.

The MCT static MAC addresses configured on a local node are advertised to remote MCT node for learning. While advertising the MAC using the MAC advertisement route, it uses the MAC mobility extended-community route to identify the MAC as static using the sticky MAC field. On the remote node, when MAC advertisement is received for a static MAC address, the sticky MAC information is saved along with the MAC entry.

When an MCT static MAC address is deleted, a MAC withdrawal route is sent to the remote peer to delete the MAC entry from its database.

When a CEP interface is down and if any static MAC entries are present, MAC Delete messages are sent to the remote node to flush the entries.

When a CCEP interface is down and if any static MAC entries are associated with the client, the MAC addresses are moved to point to the remote MCT peer. The MAC addresses are moved back to the CCEP when the interface comes back up.

On a local MCT node, when a cluster is UP and you configure a static MAC on a CEP or CCEP interface, the node synchronizes the MAC address to the remote MCT node. The remote node processes the MAC address and adds it to the FDB. On the remote MCT node, you can configure the same MAC as the static MAC address for the client 1 CCEP interface since it is configured on the same client CCEP interface. No additional static MAC configurations on the remote node are required since the same MAC are already part of the local MCT node.

When the cluster is down on the local and remote MCT nodes, both nodes are independent as clusters that can be independently configured with the static MAC addresses for the CEP or CCEP interface. However, when the cluster is brought up, the static MAC addresses are synchronized from both nodes and the addresses on the remote node are rejected since the local configuration takes precedence. The misconfiguration remains until corrected.

MAC learning

MAC learning over CEP interfaces is similar to Layer 2 learning where the MAC advertisement route is sent to the cluster peer to synchronize the learned MAC address. MAC learning over CCEP interfaces is a two-step process in which the MAC entry is added into the MDB first. The best MAC entry is chosen and installed into the FDB and the MAC advertisement route is sent to the cluster peer for synchronization of the learned MAC address.

The following rules are used for MAC learning:

- If a static MAC address is configured on the CEP port, it is learned as the Static MAC and the advertisement route message is sent to the peer. In this case, the ESI is set to NULL. On the peer MCT node, the MAC address programmed on the cluster peer is static towards the MCT peer.
- If a static MAC address is configured on the CCEP port, a MAC advertisement route is sent to the peer. In this case, the MAC entry is associated with the ESI of the MCT client. The peer MCT node programs the MAC address as static over the local CCEP interface.
- Dynamic MAC learning from the CEP is similar to basic Layer 2 MAC learning. A MAC advertisement route is sent to the peer. In this case, the ESI is set to invalid or NULL. On the peer MCT node, the MAC address programmed on the cluster peer is static towards the MCT peer.
- Dynamic MAC learning from the CCEP occurs as a CCL MAC. A MAC advertisement route is sent to the peer. In this case, the MAC entry is associated with the ESI of the

cluster client. The peer MCT node programs the MAC address as static over the local CCEP interface.

MAC aging rules

The following rules are defined for MAC aging:

- The local MAC age over CEP interface is similar to the Layer 2 MAC age. After the local MAC delete, the MAC withdrawal route is sent to the MCT peer.
- The local MAC age over CCEP interface is considered aged only if all MCT nodes age out the entry. When the MAC that ages on one of the MCT node local MDB is deleted, if the remote MDB present MAC is reprogrammed as the CCR, else the MAC is removed from the local FDB, the MAC withdrawal route is sent to MCT peer.
- The remote MAC addresses that are learned through the MAC advertisement route does not age out. They can only be removed by the MAC withdraw messages from the peer.

MAC movement

A MAC address is considered to be moved when the same MAC address is received on a different interface with same VLAN. In MCT, a MAC movement is allowed on both local and remote nodes.

The following table describes the allowed MAC movements in MCT.

Table 32: MCT MAC movement

MAC movement scenario	Behavior
Local dynamic MAC move from CEP1 to the CEP2 edge interface on MCT1.	On local node MCT1, the MAC address is updated to point to the new CEP2 interface. There is no MAC route update required to the remote MCT node. As on the remote node, the MAC always point towards the MCT peer for MAC addresses.
Local dynamic MAC move from CEP1 edge interface to the CCEP1 client interface on MCT1.	On local node MCT1, the MAC address is updated to point to the new client interface CCEP1. A MAC update route is sent with the new ESI of client 1. The remote node updates the MAC address to point to the CCEP of client 1.
CCEP1 interface (client 1) to CCEP2 interface (client 2) on MCT1.	On local node MCT1, the MAC address is updated to point to the new client interface CCEP2. A MAC update is sent with the new ESI of client 2 to the remote node. The remote node updates the MAC address to point to the CCEP of client 2.
Local dynamic MAC move from CCEP1 interface (client 1) to CEP1 edge interface on MCT1.	On local node MCT1, the MAC address is updated to point to the new edge interface CEP1. A MAC update is sent with the new ESI 0 to the remote node. The remote node MCT2 updates the MAC address pointing to the MCT1 node.

Table 32: MCT MAC movement (continued)

MAC movement scenario	Behavior
For a MAC learned on a CEP port locally (MCT1). Dynamic MAC move to a CEP port on the remote node (MCT2)	On the MCT2 node for the MAC learned from MCT1, it is considered as a MAC move when it is learned on a CEP port. The MAC is updated as local on MCT2 and now points to the Dynamic on the CEP port on MCT2 instead of pointing to MCT1 node MCT2 sends an updated MAC to MCT1. MCT1 updates the MAC as remote and points to the MCT2 .
For a MAC learned on a CEP port locally (MCT1). Dynamic MAC move to CCEP1 on MCT2.	On the MCT2 node for the MAC learned from MCT1, it is considered as a MAC move when the same MAC is learned on a CCEP1 port. The MAC is updated as CCL on MCT2 and now points to the local CCEP1 port on MCT2 instead of pointing to the MCT1 node. MCT2 sends a CCL MAC updated to MCT1. MCT1 updates the MAC as CCR and point to the CCEP1 port.
For a MAC CCL learned on a CCEP1 port locally (MCT1). Dynamic MAC move to CCEP2 on remote MCT2 node.	On the MCT2 node for the CCR MAC learned from MCT1 for client 1, it is considered as a MAC move when the same MAC is learned on client 2 over the CCEP2 port. The MAC is updated as CCL on MCT2 and now points to the local CCEP2 port on MCT2 instead of pointing to CCEP1. From MCT2, it sends a CCL MAC updated to MCT1. MCT1 updates the MAC as CCR and points to the CCEP2 port.
For a MAC CCL learned on a CCEP1 port locally (MCT1). Dynamic MAC move to CEP port on remote MCT2 node.	On the MCT2 node for the CCR MAC learned from MCT1 for client1, it is considered as a MAC move when the same MAC is learned on the CEP port. The MAC is updated as Dynamic on MCT2 and points to the local CEP port on MCT2 instead of pointing to CCEP1. From MCT2, it sends the MAC updated to MCT1. MCT1 updates the MAC and points to the MCT2.

MAC address deletion

The following rules are defined for MAC address deletion. Note that every deletion triggers the MAC resolution algorithm and reprograms the MAC entry if required.

- If the CEP interface is down, MAC addresses are deleted locally and individual MAC deletion messages are sent to the peer.
- If the CCEP local port is down and the remote CCEP is down, MAC addresses are deleted locally and the ESI withdraw message is sent to the MCT peer instead of sending individual MAC delete messages.
- If the CCEP local port is down and the remote CCEP is up, all local MAC addresses are moved to point to the remote MCT peer including the static MAC addresses associated with the CCEP.
- When the client entry is undeployed, all MAC addresses are deleted locally, and the ESI withdraw message is sent to the MCT peer to delete all associated client MAC addresses.

MCT configuration considerations

General considerations

- MCT Peers need to run the same version of SLX-OS.
- MCT does not support any variant of Spanning Tree Protocol (STP) on ICL links, cluster client ports, or any VLAN that is part of the cluster. STP is disabled by default.

  Note
  If you enable STP on MCT nodes, each node acts as an independent (not interconnected) STP switch. This situation results in STP state flap at the node connected to the CCEP port, because it receives two different Bridge Protocol Data Units (BPDU) on that CCEP port

- SLX-OS supports dynamic and static LAG between the MCT node and client.
- SLX MCT configurations (Layer 2 or Layer 3) do not require the Advance Feature license.

Peer considerations

- For both MCT peers, the MCT peer IP address must match the source IP address of the peer node.
- The source IP and the peer IP should be in the same subnet.
- You must configure the same client ID on both MCT peers for the link or CCEP that is connected to the same client.
- SLX MCT peers must be physically connected to each other.

VLAN considerations

- If you configure an MCT port channel for multiple VLAN or VE interfaces, use static routes instead of OSPF ECMP.
  - As a best practice, configure a static route for MCT peers through the MCT VLAN (VE 10 in this example). A static route has a lower administrative distance than OSPF and places only one route in the routing table.

LACP considerations

- A common system ID is used on both MCT nodes. This is a fixed value.
- The LACP fields have the following settings:

```
Key = MCT_LACP_KEY_BASE (3000) + client_ID

Port ID (16-bit unique value) =
5-bit (slot value) + 8-bit (port value + 3-bit) (MCT position offset)
```

Configuring MCT

Ensure that the following configurations exist:

- Layer 3 interface for the cluster peer interface
- VLANs and bridge domains function as the MCT members
- Port channel for Link Aggregation or an Ethernet interface as a client interface

Perform the following steps.

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

```
device# configure terminal
```

2. Create a cluster on the device.

```
device(config)# cluster leaf1_2
```

3. Configure the peer IP address.

```
device(config-cluster-leaf1_2)# peer 40.1.1.2
```

   The peer IP address is the IP address configured on the peer switch for the peer-interface.

4. Configure the peer interface.

```
device(config-cluster-leaf1_2)# peer-interface port-channel 40
```

   The peer interface should be a valid Layer 3 interface.

5. Configure the port channel interface.

```
device(config-cluster-leaf1_2)# exit
device(config)# int port-channel 40
```

6. Configure port-channel IP address.

```

device(config-port-channel-40)# ip address 40.1.1.1/24
```

7. Configure the port channel range.

```
device(config-Port-channel-40)# exit
device(config)# cluster leaf1_2
device(config-cluster-leaf1_2)# int port-channel 1-2,5
```

   This allows you to replicate a single command across multiple interfaces.

8. Set the cluster client ID for the port-channel (auto allocated ID in this case).

```
device(config-Port-channel-1-2,5)# cluster-client auto
```

The following example is the steps in the previous configuration.

```
device# configure terminal
Entering configuration mode terminal
device(config)# cluster leaf1_2
device(config-cluster-leaf1_2)# peer 40.1.1.2
device(config-cluster-leaf1_2)# peer-interface port-channel 40
device(config-cluster-leaf1_2)# int port-channel 40
device(config-port-channel-40)# ip address 40.1.1.1/24
device(config-cluster-leaf1_2)# int port-channel 1-2,5
device(config-Port-channel-1-2,5)# cluster-client auto
device(config-Port-channel-1-2,5)# exit
device(config)#
```

Taking the MCT node offline for maintenance

If you need to take an MCT device offline for maintenance or an upgrade, perform the following steps to minimize traffic loss.

1. Verify the state of the MCT node and it's peer using **show cluster**.

```
device# show cluster MCT
Cluster MCT
==============
Cluster State: Active
Bringup Delay: 90 seconds
Configured Member Vlan Range: All
Active Member Vlan Range: 1-4090
Configured Member BD Range: All
Active Member BD Range: 10,1001-3000
No. of Clients: 4

 Peer Info:
 ==========
 Peer IP: 15.1.1.1, State: Up
 Peer Interface: Port-channel 256, Source IP: 15.1.1.2

 Keep-Alive:
 ==========
 IP: 10.20.232.52, State: Up
 Interface: Management 0 (mgmt-vrf), Source IP: 10.20.233.224
 Client-Isolation Role: Primary

 Client Info:
 ============
 Interface               Id               Description          Local/Remote State
Exceptions
 ---------               --               -----------          ------------------
----------
 Port-channel 1          1001                                  Up   /
Up
 Port-channel 2          1002                                  Up   /
Up
 Tunnel 32771            42.42.42.42      VxLAN                Up   /
Up
 PW                      34816            VPLS/VLL             Down / Down
Client Down
```

2. Enter config mode and enable system maintenance.

```
device# config
device(config)# system maintenance
device(config-system-maintenance)# enable
```

Note
Do not write the configuration changes made in the previous steps to the startup-configuration file.

To bring the MCT node back online, perform one of the following actions.

- If you upgraded or downgraded the device, select the **coldboot** option under the firmware download menu.
- For any other reason, execute the **reload system** command. Since the changed configuration was not saved, the reload reverts the configuration changes that had taken the MCT node offline.

Configuring additional MCT cluster parameters

Peer Keepalive

The peer-keepalive feature helps in distinguishing interface failure and cluster peer node failure, and takes action accordingly. An out-of-band keep-alive session is recommended between the nodes. This helps to distinguish peer-interface failures from peer node reboot. Different actions can then be taken for these peer-interface failures.

The keep-alive session has 2 flavors:

- Auto keep-alive session (default)
  - Does not require any explicit user configuration, and is run in the management VRF. The management IPs of the two nodes are exchanged once the cluster session is established and an out-of-band keep-alive session is started between them.
- Manual keep-alive session
  - A source interface with IP and destination IP can be specified to be used for keep-alive session. This can be in any VRF. These IP addresses cannot match the peer IP and source IP used for the CCP session to ensure proper out-of-band functionality.

The **peer-keepalive role** determines the node's behavior when the ICL goes down, but the MCT peer remains up.

- Use **auto** to assign roles automatically. By default, node with lower IP is selected as primary and higher IP is selected as secondary. This behavior can be changed by manually assigning the primary/secondary role under the cluster.
- The primary role is responsible for forwarding all traffic when the ICL goes down. It keeps all the cluster-client ports up and becomes the Designated Forwarder for all VLANs/BDs on the tunnels and PWs.
- The secondary role isolates the secondary node, brings down all client interfaces and tunnels, and isolates all client traffic. this behavior is only effective when the keep-alive session is up while CCP session goes down. In all other cases, primary and secondary will not have an impact on functionality.

When a keep-alive session goes down but the CCP session remains up, the cluster continues to operate in regular mode. No traffic flow change is observed. This mode is not recommended because there is no peer-link protection. If the CCP session were also to go down, the behavior would be same as a peer reload; both nodes would keep the clients up and become the DF for all VLANs/BDs.

Use the **peer-keepalive** command to configure the peer-keepalive mode on a node, as shown in the following example.

```
device(config)# cluster leaf1_2
device(config-cluster-leaf1_2)# peer-keepalive
device(config-peer-keepalive)# role primary
device(config-peer-keepalive)#
```

Configuring peer-keepalive destination

Use the **peer-keepalive destination ip** command to overwrite the default keepalive session with the configured session between the specified source and the destination IP address

```
device(config)# cluster leaf1_2
device(config-cluster-leaf1_2)# peer-keepalive destination 10.20.161.158
device(config-cluster-leaf1_2)#
```

Moving the traffic from an MCT node to the remote node

Use the **shutdown {all | clients}** command to move all the traffic on the node to the remote MCT node by disabling the local client interfaces administratively. Use of the all parameter will shutdown the MCT peer and clients.

```
device(config)# cluster leaf1_2
device(config-cluster-leaf1_2)# shutdown clients
device(config-cluster-leaf1_2)#
```

Displaying MCT information

You can display detailed MCT information and related MCT MAC addresses.

Displaying the cluster information

The following example shows the information of the cluster on the SLX-OS device.

```
device# show cluster

Cluster S1-A1
==============
Cluster State: Active
Bringup Delay: 90 seconds
Configured Member Vlan Range: All
Active Member Vlan Range: 1-4090
Configured Member BD Range: All
Active Member BD Range: 1-500
No. of Clients: 9

 Peer Info:
 ----------
 Peer IP: 12.0.0.2, State: Up
 Peer Interface: Port-channel 1, Source IP: 12.0.0.1

 Keep-Alive:
 ==========
 IP: 10.20.161.158, State: Up
 Interface: Management 0 (mgmt-vrf), Source IP: 10.20.161.185
 Client-Isolation Role: Secondary

 Client Info:
 ============
 Interface               Id               Description          Local/Remote State
Exceptions
 ---------               --               -----------          ------------------
----------
 Port-channel 11         1                towards Fusion-1 1   Up   /
Up
```

```
 Port-channel 12         2                towards Fusion-2 1   Up   /
Up
 Port-channel 13         3                towards Avalanche-2  Up   /
Up
 Port-channel 14         4                towards Fusion-1 2   Up   /
Up
 Port-channel 15         5                towards Aval-2 2     Up   /
Up
 Port-channel 16         6                towards MLXe8        Up   /
Up
 Port-channel 17         7                towards MLXe4        Up   /
Up
 Port-channel 18         8                towards NH-9150      Down / Down
Client Down                                                      n
 Port-channel 19         9                towards MLXe8 Static Up   / Up
```

Displaying the cluster client information

The following example displays all client information for cluster 1.

```
device# show cluster client 22
Client Info:
------------
Interface: Port-channel 20, client-id: 1020, Deployed, State: Up
Description          : U27
Local Vlans          : 1,61-300
Local Bridge Domains    : 20,100
Remote Vlans         : 1,61-300
Remote Bridge Domains      : 20
DF Elected for all vlans/BDs
```

The following example displays cluster client-pw information.

```
device# show cluster client-pw
Client Info:
------------
Interface: PW, client-id: 34816, Deployed, State: Down
Description:
Local Vlans:
Local Bridge Domains          : 20
Remote Vlans                  :
Remote Bridge Domains         : 20
Number of DF Vlans            : 0
Elected DF for vlans          :
Number of DF Bridge Domains   : 1
Elected DF for Bridge Domains : 20
```

The following example displays cluster client IP.

```
device# show cluster client 30.30.30.30
Client Info:
------------
Interface: Tunnel 61441, client-id: 30.30.30.30, Deployed, State: Up
Description:
Local Vlans                   : 41-60,400
Local Bridge Domains          : 20
Remote Vlans                  : 400
Remote Bridge Domains         : 20
Number of DF Vlans            : 20
Elected DF for vlans          : 41-60
Number of DF Bridge Domains   : 0
Elected DF for Bridge Domains :
```

Displaying member VLAN information

The following example displays the member VLAN information for the cluster.

```
device# show cluster member vlan
VLAN-ID   VNI-ID              Forwarding state
-------   ------              -------------------
  1          1                   Up
  40         40                  Up
  41         41                  Up
  42         42                  Up
```

Displaying and clearing the MAC address table cluster information

The following example shows how to display the MCT cluster information in the MAC address table.

```
device# show mac-address-table cluster leaf1_2
Vlan/BD'Id Mac-address     Type        State    Ports
100(V)     0010.a111.aaaa  CCL         Active   ETH 0/1
100(V)     0010.a111.aa22  Static-CCL  Active   ETH 0/1
100(V)     0010.a111.bbbb  CCR         Active   ETH 0/1
200(V)     003d.a111.1111  Dynamic     Active   Eth 0/2
200(V)     003d.a111.1122  Static      Active   Eth 0/2
```

The MAC Type for an MCT cluster displays the following information:

- For the client MAC behavior, MAC addresses are learned as CCL on the local MCT node and CCR on the remote MCT node pointing to the CCEP interface.
- For static MAC addresses over client interfaces, Static-CCL and CCR are displayed.

You can also view the MAC entries for a specific client.

Clearing the MCT cluster MAC table entries

You can clear all cluster entries from the MAC address table or the entries for a specified client. The following example clears the MAC entries for client 3 of cluster leaf1_2.

```
device# clear mac-address-table cluster leaf1_2 client 3
```

When the remote MCT peer receives the MAC withdrawal message, it only deletes the remote MAC entry. To clear MAC addresses on both nodes, you must issue **clear mac-address-table** commands on both MCT nodes.

VPLS and VLL MCT on the SLX 9640 and 9540 devices

Note
For more information on VPLS and VLL, refer to the "VPLS and VLL Layer 2 VPN services" chapter.

For VPLS MCT, a point-to-multipoint (p2mp) bridge domain is added to the MCT cluster. For VLL MCT, a point-to-point (p2p) bridge domain is added to the MCT cluster. The VPLS or VLL horizon is added as a pseudowire (PW) client.

VPLS/VLL MCT supports PW redundancy. At any point of time, one active-active PW path exists to reach the destination. The node on which the PW is active is called the

active node. The endpoint traffic coming from the standby node traverses through the MCT ICL session to the active node for that instance and the active MCT node takes care of the forwarding to the remote VPLS or VLL peer.

Note
For SLX-OS, the MCT cluster requires both nodes to be on SLX-OS devices. However, an SLX-OS MCT cluster that connects to a Extreme MLX cluster through VPLS or VLL is supported.

Control plane for VPLS or VLL MCT

The bridge domain is mapped to a CCP instance. For each BD, the default CCP ID is the BD ID plus 4,096. A user configured CCP ID is not supported. The CCP ID for the VLAN is the VLAN ID.

For VPLS or VLL MCT, the physical and LAG CCEP operate in active/active multi-homing mode. However, the PW operates in active/standby mode.

The designated forwarder (DF) state of the PW-Client ID represents the active PW node state for a VPLS or VLL instance. The DF election process for the PW ESI is the same as the Layer 2 process. However, VPLS and VLL MCT in the following dynamic cases do not change their role, and are driven through the PW horizon client.

- If the local endpoint is down, the remote endpoint is up.
- If the active-active PW is down on the active node, the PW on the standby node will not change to active-active.

PW redundancy for VPLS and VLL MCT

Status TLV support is enabled through a VLL (only) instance if one of the following is true.

- VLL is configured with two remote peers.
- The VLL endpoint is a MCT client CCEP port.

To support PW redundancy, configure two VLL peers under one VLL instance. One PW is for each VLL peer. Among these PWs, an active-active PW is selected and used for traffic flow to the remote side. An active-active PW is selected based on the local and remote PW redundancy preferences. A remote PW redundancy preference is received by the PW status TLV. When the bit is set, it indicates PW forwarding standby. When the bit is cleared, it indicates PW forwarding active.

PW state in VPLS or VLL MCT

The PW state in VPLS or VLL MCT is controlled by two entities. The MCT module controls its MCT state. The PW remote peer provides its PW redundancy state.

Together, they decide the operational (forwarding) state of the PW. The following table shows the PW state decisions.

MCT state	PW remote state	Operational state	PW signaling state
DF	Active	Active	Active
DF	Standby	Standby	Active
Non-DF	Any	Standby	Standby

When the PW is in active operational state, the data plane objects (such as LIF or MGID, or cross-connect for VLL) is created and be programmed into the hardware. When the PW is in standby operational state, the data plane is programmed as if this PW is down.

Note
The SLX-OS PW state table is the same as the Extreme NetIron VPLS-MCT or VLL-MCT PW state table to ensure compatibility when facing an MLX MCT cluster over a VPLS or VLL connection.

VLL-MCT data plane

A topology for VLL MCT is provided in the following figure. Two MCT clusters face each other and four PWs connect the clusters.

Note
This topology is for only one BD. Since the DF state is selected per BD, another BD can use a different PE as the active node.

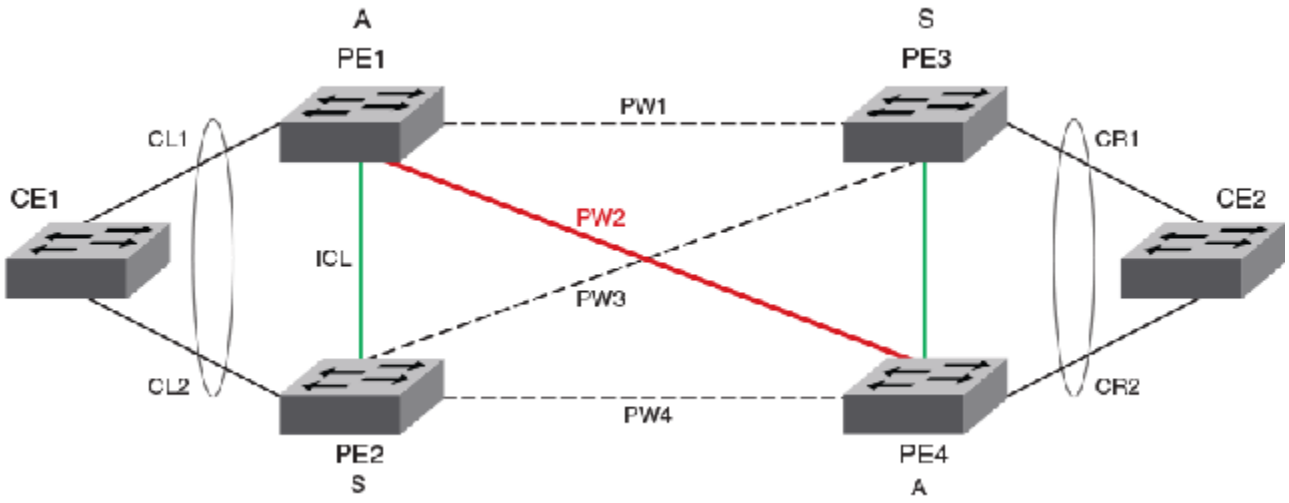


Figure 11: VLL MCT topology

When VLL MCT is activated, only one PW is operationally active between the MCT clusters, as represented by the solid line. The standby PWs are represented by the dotted lines.

The ICL link between the MCT nodes is a VxLAN tunnel.

Note
VLL MCT does not use MAC learning. BUM traffic handling is not required. It uses cross-connect instead of VSI.

Steady state traffic

Based on the previous figure, the steady state traffic is as follows:

CE1->PE1->**PW2**->PE4->CE2
CE1->PE2->ICL[Split Horizon PW or ICL]->PE1->**PW2**->PE4->CE2

Client Link down protection

When the client link (CL1) is down, the device does not change the MCT status for this VLL. Traffic from the client will be received on CL2 to PE2 and forwarded using the VxLAN tunnel from PE2 to PE1. The traffic flow from the client is as follows:

CE1 -> PE2 -> [Split Horizon PW or ICL] -> PE1 -> **PW2** -> PE4 -> CE2

Active MCT Node protection

VLL MCT provides protection when one PE node has a failure including a software or hardware failure, or a power down. In the case when the active MCT node (PE1) is down, the standby MCT node acts as active and uses corresponding PWs for the traffic flow from the client. The traffic flow from the client is as follows:

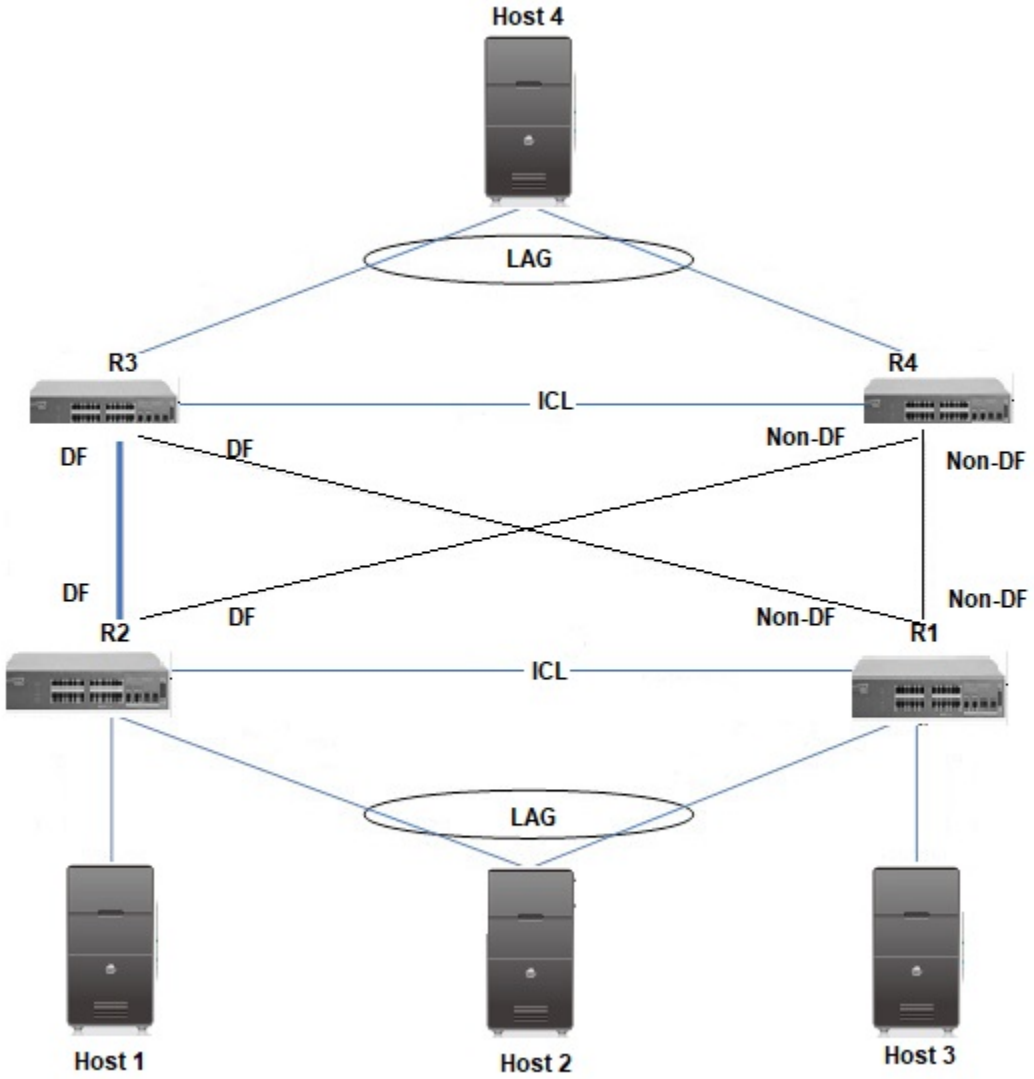
CE1 -> PE2 -> **PW4** -> PE4 -> CE2

Note
For active-active PW link protection when the PW redundancy status changes, the device relies on the MPLS configuration. There should not be a case where PW2 is down and PW4 is up. The configuration ensures that both PW2 and PW4 are UP or DOWN. When PW2 is not active-active due to a role change on PE4, PW1 will become active-active.

VPLS-MCT data plane

The main case topology for VPLS MCT is provided in the following figure. Two MCT clusters face each other and four PWs connect the clusters.

Note
This topology is for only one BD. Since the DF state is selected per BD, another BD can use a different PE as the active node.

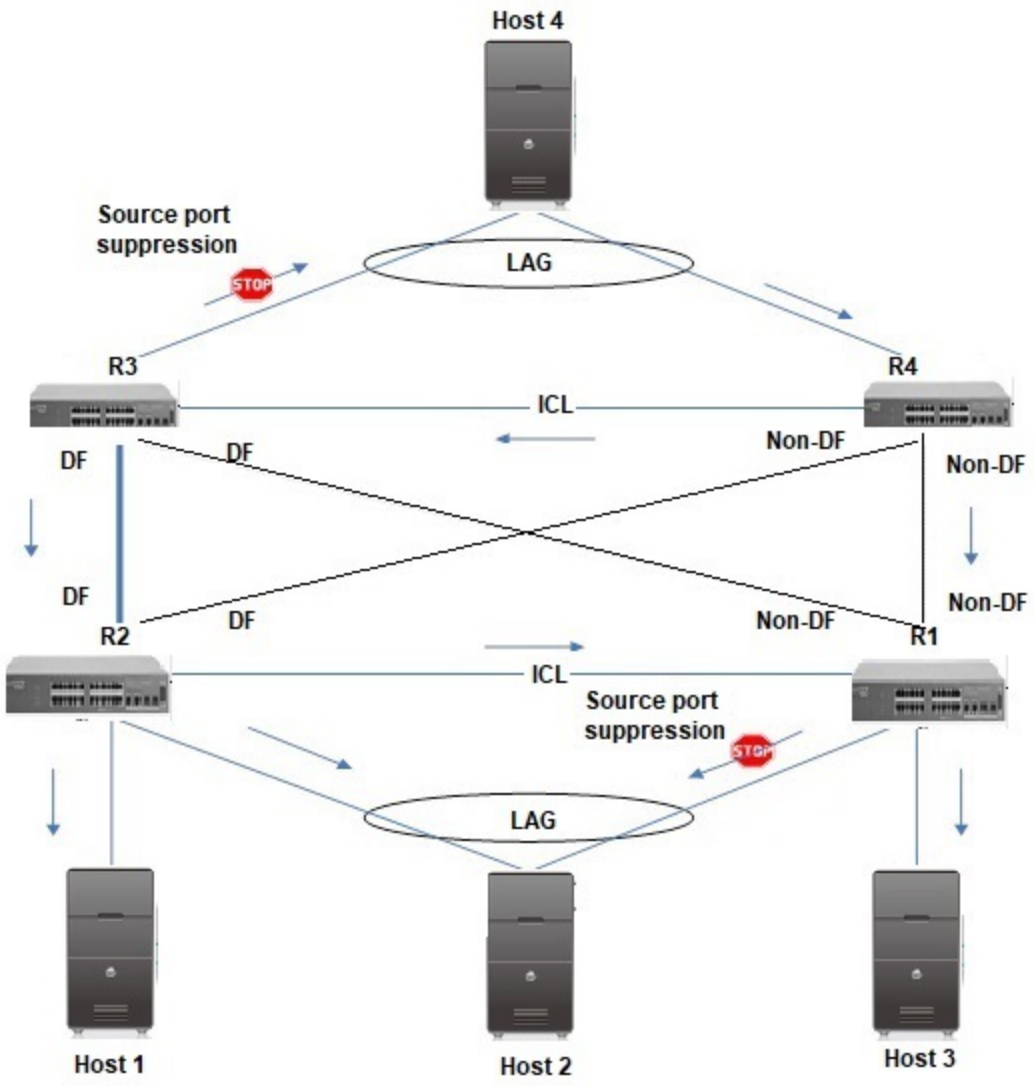


When VPLS MCT is activated, only one PW is operationally active between the MCT clusters, as represented by the solid line. The operational standby PW is represented by the dotted lines.

The ICL link between the MCT nodes is a VxLAN tunnel.

VPLS MCT BUM traffic

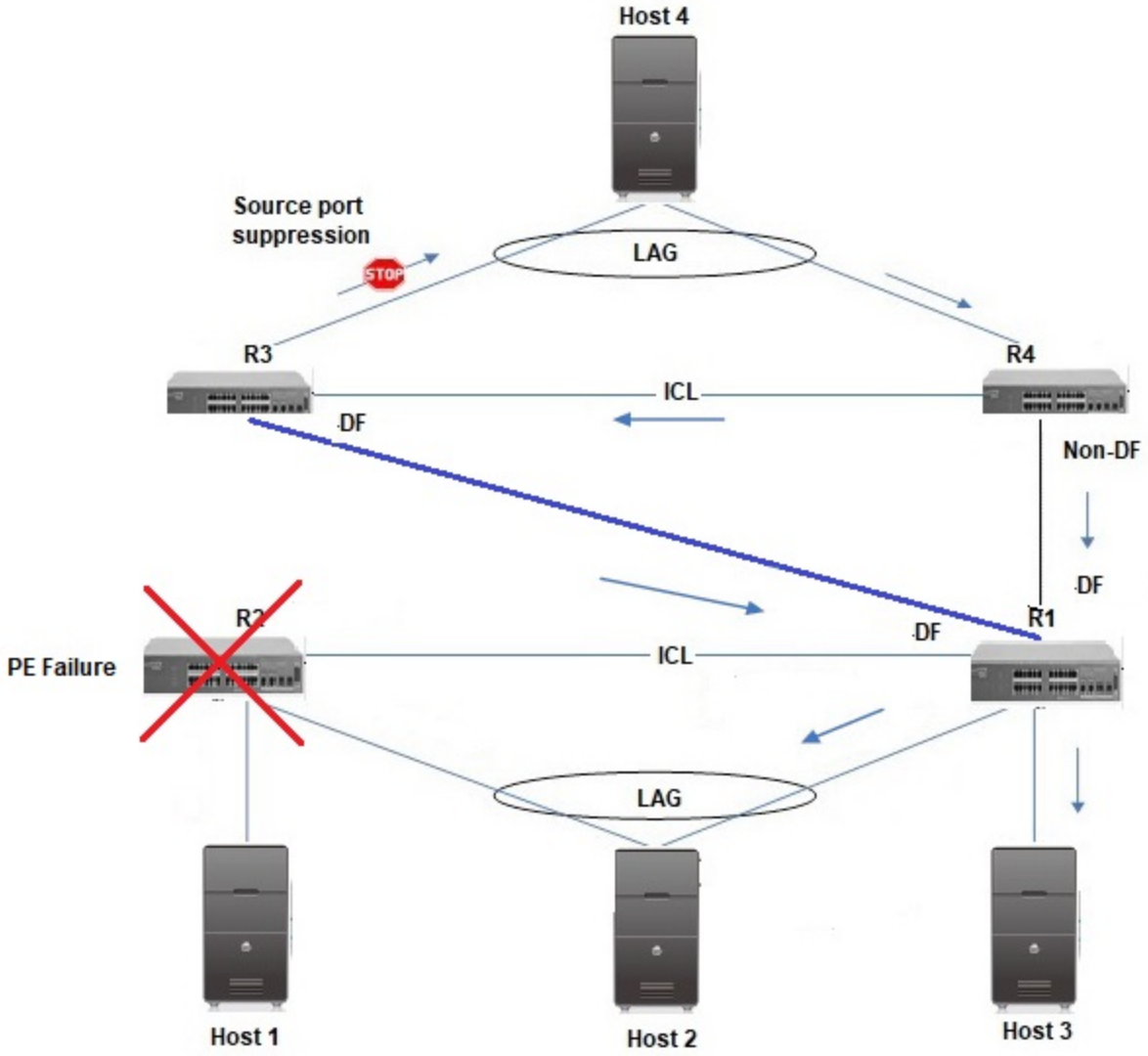
The following figure illustrates how a BUM packet that starts from host 4 travels through the VPLS-MCT network to reach hosts 1, 2, and 3.



VPLS-MCT PE node protection

VPLS MCT provides protection when one PE node has a failure, including a software or hardware failure, or a power-down event. When the active PE fails in MCT, the standby PE becomes the active PE and all PWs on this node transits into MCT active state.

The following figure shows the BUM packet flow after an MCT PE switch-over event.



Note
VPLS MCT does not support PW link protection.

VPLS MAC Management

In VPLS-MCT topologies, MAC addresses can be learned either on a local PW or from the remote MCT node. The same MAC can be learned locally or remotely, but not from both MCT nodes.

For MCT remote MAC addresses, aging is disabled, and they can only be deleted when delete notifications are received from the remote MCT node that previously advertised it for learning.

The following terminologies are associated with MCT MAC entries.

- Dynamic—MAC addresses learned locally on CEP ports
- Cluster Remote (CR)—MAC addresses learned on remote CEP ports
- Cluster Client Local (CCL)—MAC addresses learned locally on a client interface
- Cluster Client Remote (CCR)—MAC addresses learned on a remote client interface

VPLS MAC addresses that are learned locally are classified as Dynamic, and the corresponding VPLS MAC addresses that are learned remotely are classified as Cluster Remote (CR).

Static MAC handling

Static MAC configuration over the local VPLS endpoints is supported. Static MAC pointing to the PW that is established with the remote VPLS PE is not supported.

MAC learning

MAC addresses learned from the PW on the active PE triggers CR MAC synchronization messages that are sent to the peer. The PW ESI is used in this MAC route. VPLS CR MAC addresses point to the active MCT node since no local forwarding path on the standby PE traffic is expected to be switched by the active MCT node.

MAC aging

When the VPLS MAC ages on the active node, the MAC address is locally flushed and the CR MAC withdrawal route is sent to remote MCT node to flush the MAC.

VPLS MAC movement

A MAC address is considered to be moved when the same MAC address is received on a different interface with same VLAN. In MCT, a MAC movement is allowed on both local and remote nodes.

The following table describes the allowed VPLS MAC movements in MCT.

MAC movement scenario	Behavior
Local dynamic MAC move from PW A to PW B on MCT1.	On local node MCT1, the MAC address is updated to point to the new PW interface. There is no MAC route update required to the remote MCT node. As on the remote node, the MAC always point towards the MCT peer for all VPLS addresses.
Local dynamic MAC move from PW to the Layer 2 CCEP1 client interface on MCT1.	On local node MCT1, the MAC address is updated to point to the new client interface CCEP1. A MAC update route is sent with the new ESI of client 1. The remote node updates the MAC address to point to the CCEP of client 1.

MAC movement scenario	Behavior
For a MAC learned on a PW locally (MCT1). Dynamic MAC move to CCEP1 on MCT2.	On the MCT2 node for the CR MAC learned from MCT1, it is considered as a MAC move when the same mac is learned on a CCEP1 port. The MAC is updated as CCL on MCT2 and now points to the local CCEP1 port on MCT2 instead of pointing to the MCT1 (PW) node. MCT2 sends a CCL MAC update to MCT1. MCT1 updates the MAC as CCR and points to the CCEP1 port.
For a MAC CCL learned on a CCEP1 port locally (MCT1). Dynamic MAC move to the PW on the remote MCT2 node.	On the MCT2 node for the CCR MAC learned from MCT1 for client 1, it is considered as a MAC move when the same MAC is learned on the PW. The MAC is updated as the Dynamic on MCT2 and now points to the PW on MCT2 instead of pointing to CCEP1. From MCT2, it sends a Dynamic MAC update to MCT1. MCT1 updates the MAC to point to MCT2.
Local dynamic MAC move from PW (MCT1) to CEP1 client interface on MCT1.	On local node MCT1, the MAC address is updated to point to the new interface CEP1. A MAC advertise route is sent with ESI 0 to the remote MCT node. The remote node MCT2 updates the MAC address to point to the MCT1 node.
For a MAC learned as Dynamic on a PW locally (MCT1). Dynamic MAC move to CEP on MCT2.	On the MCT2 node for the CR MAC learned from MCT1, it is considered as a MAC move when the same mac is learned on the CEP port. The MAC is updated as Dynamic on MCT2 and points to the local CEP1 port. From MCT2, it sends a Dynamic MAC updated to MCT1. MCT1 updates the MAC as EVPN and points to the MCT2 node.
For a MAC learned as Dynamic on a CEP1 port locally (MCT1). Dynamic MAC move to PW on remote MCT2 node.	On MCT2 node for the CR MAC learned from MCT1 for CEP, it will be considered as a MAC move when the same mac is learned on the PW. The MAC is updated as Dynamic on MCT2 and now points to the PW on MCT2. MCT2 sends a Dynamic MAC updated to MCT1 with the ESI of the PW client, MCT1 now should updated the MAC point to the MCT2.

MAC address deletion

The following rules are defined for MAC address deletion. Every MAC deletion triggers the MAC resolution algorithm and reprograms the MAC entry if required.

- If a PW is down, MAC addresses flushed locally and individual MAC deletion messages are sent to the MCT Peer. This is similar to the Layer 2 CEP port-down handling.
- If PW client is undeploy on MCT 1, only one MAC withdraw message is send to MCT 2.

  All MAC addresses tagged to the PW client are flushed.
- If MCT 2 detects that MCT 1 is down or if the session is down, all VPLS MAC addresses learned from MCT 1 are flushed.

Configuration Considerations and Limitations for VPLS and VLL MCT

- Hitless ISSU is not supported. Before starting ISSU, issue the **shutdown clients** command on the PE where the ISSU is planned to gracefully move the traffic to the MCT peer. Similarly, use the **force-standby** or **no deploy** command for the PW CCEP before starting ISSU.
- Statistics are not supported.
- For VPLS MCT, consider the following:
  - Configuring a cluster peer as a BD peer impacts data traffic.
  - You can use the **shutdown clients** command to shutdown traffic on one node before a software upgrade. After you issue this command, all PWs are put into standby mode.
  - **shutdown clients** brings down all CCEP interfaces. Other nodes attempt client-isolation logic, and you may see the Strict behavior.
  - Logical-interface shutdown brings down the admin state of the CCEP LIF interface if the parent port is an MCT client interface and does not trigger a DF re-election.
  - Protection for a PW link failure is not supported. Active forwarding paths does not occur between the nodes.
- For VLL MCT, consider the following:
  - Cross-connect is used instead of VSI.
  - MAC learning is not required.
  - BUM traffic handling is not required.
  - The MCT role does not depends upon endpoint as well as the PW redundancy status.

Configuring MCT for VPLS or VLL

- Before configuring VPLS MCT, configure a point-to-multipoint (p2mp) bridge domain.
- Before configuring VLL MCT, configure a point-to-point (p2p) bridge domain.

For information on configuring VLL or VPLS bridge domains, refer to the "VPLS and VLL Layer 2 VPN services" chapter.

For information on configuring the MCT cluster and client, refer to Configuring MCT on page 107. Their full configuration is provided in the example after the steps.

Perform the following steps to configure MCT for VPLS or VLL.

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

```
device# configure terminal
```

2. Access the cluster on the device.

```
device(config)# cluster leaf1_2
```

3. Configure the peer IP address.

```
device(config-cluster-leaf1_2)# peer 40.1.1.2
```

4. Configure the peer interface port-channel.

```
device(config-cluster-leaf1_2)# peer-interface port-channel 40
```

5. Create the PW client for the cluster.

```
device(config-cluster-leaf1_2)# client-pw
```

   Only one instance of the PW client represents all VPLS or VLL PWs over all bridge domains.

6. Configure the member bridge-domain for MCT.

```
device(config-cluster-leaf1_2)# member bridge-domain all
```

   By default the member bridge-domain all is configured.

7. Configure the interface port-channel.

```
device(config-cluster-leaf1_2)# int port-channel 40
```

8. Set the IP address and subnet mask for the port-channel interface.

```
device(config-Port-channel-40)# ip address 40.1.1.1/24
```

9. Configure the port-channel interface range?

```
device(config-Port-channel-40)# int port-channel 1-2,5
```

10. Set the cluster client to auto.

```
device(config-Port-channel-1-2,5)# cluster-client auto
```

The following example are the steps in the previous configuration with the additional configuration of the MCT cluster and client.

```
device# configure terminal
device(config)# cluster leaf1_2
device(config-cluster-leaf1_2)# peer 40.1.1.2
device(config-cluster-leaf1_2)# peer-interface port-channel 40
device(config-cluster-leaf1_2)# client-pw
device(config-cluster-leaf1_2)# member bridge-domain all
device(config-cluster-leaf1_2)# int port-channel 40
device(config-Port-channel-40)# ip address 40.1.1.1/24
device(config-Port-channel-40)# int port-channel 1-2,5
device(config-Port-channel-1-2,5)# cluster-client auto
device(config-Port-channel-1-2,5)#
```

Displaying information related to VPLS and VLL MCT

Displaying PW client information on an MCT cluster

The following example displays the configuration and state information of the PW client on the MCT cluster.

```
device# show cluster
Cluster c1
================
Cluster State: Active
Bringup Delay: 90 seconds
Configured Member Vlan Range: All
Active Member Vlan Range: 1
Configured Member BD Range: All
Active Member BD Range: 501
No. of Clients: 2


 Peer Info:
```

```
 ==========
 Peer IP: 4.1.1.10, State: Up
 Peer Interface: Ethernet 0/4, Source IP: 4.1.1.11


 Keep-Alive:
 ==========
 IP: 10.24.8.220, State: Up
 Interface: Management 0 (mgmt-vrf), Source IP: 10.24.8.221
 Client-Isolation Role: Primary


 Client Info:
 ============
 Interface               Id               Description          Local/Remote State
Exceptions
 ---------               --               -----------          ------------------
----------
 Port-channel 10         1010                                  Up   /
Up
 PW                      34816                                 Up   /
Up
```

The following example displays only PW client and its bridge-domain information on the MCT cluster.

```
device# show cluster client-pw

Client Info:
============
Interface: PW, client-id: 34816, Deployed, State: Up
Description:
Local Vlans:
Local Bridge Domains  : 501
Remote Vlans          :
Remote Bridge Domains : 501
Number of DF Vlans    : 0
Elected DF for vlans  :
Number of DF Bridge Domains   : 1
Elected DF for Bridge Domains : 501
```

The following example displays the multicast and unicast labels, and forwarding state for the cluster member bridge domain.

```
device# show cluster member bridge-domain
BD-ID       VNI-ID              Forwarding state
-------     ------              -------------------
  501        4597                Up
```

Displaying the MCT state on a bridge domain

In the **show bridge-domain** output, the MCT Enabled field displays whether the bridge domain is configured under a cluster configuration.

```
device# show bridge-domain 501
Bridge-domain 501
-------------------------------
Bridge-domain Type: MP, VC-ID: 501
 MCT Enabled: TRUE
Description:
Number of configured end-points: 3, Number of Active end-points: 3
VE id: N/A, if-indx: N/A, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: default, mac-limit: 0
VLAN: 501
```

```
  Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
  Tagged ports: po10.501
  Un-tagged ports:
VNI: 4597
  Tunnels: 1(1 up)
  Tunnels: tu32769.4597
Total VPLS peers: 1 (1 Operational):

VC id: 501, Peer address: 9.9.9.9, State: Operational, uptime: 12 min 44 sec
Load-balance: False, Cos Enabled: False,
Tunnel cnt: 1
rsvp to_a9 (cos_enable:False cos_value:0)
Assigned LSPs count:0 Assigned LSPs:
Local VC lbl: 800769, Remote VC lbl: 866325,
Local VC MTU: 1500, Remote VC MTU: 1500,
Local VC-Type: 5, Remote VC-Type: 5
Local PW preferential Status: Active, Remote PW preferential Status: Active
Local Flow Label Tx: Disabled, Local Flow Label Rx: Disabled
Remote Flow Label Tx: Disabled, Remote Flow Label Rx: Disabled
Local Control Word: Disabled, Remote Control Word: Disabled
```

Displaying MAC address information for a VPLS bridge domain on an MCT cluster

The following example displays the MAC address table for bridge domain of an MCT cluster.

```
device# show mac-address-table bridge-domain
Type Code - CCL:Cluster Client Local MAC
            CCR:Cluster Client Remote MAC
            CR:Cluster Remote MAC
VlanId/BDId   Mac-address       Type        State      Ports/LIF/PW/T
501 (B)       0000.0500.0200    Dynamic     Active     9.9.9.9
501 (B)       0000.0500.0500    Dynamic-CCL Active     Po 10.501
Total MAC addresses    :  2
```

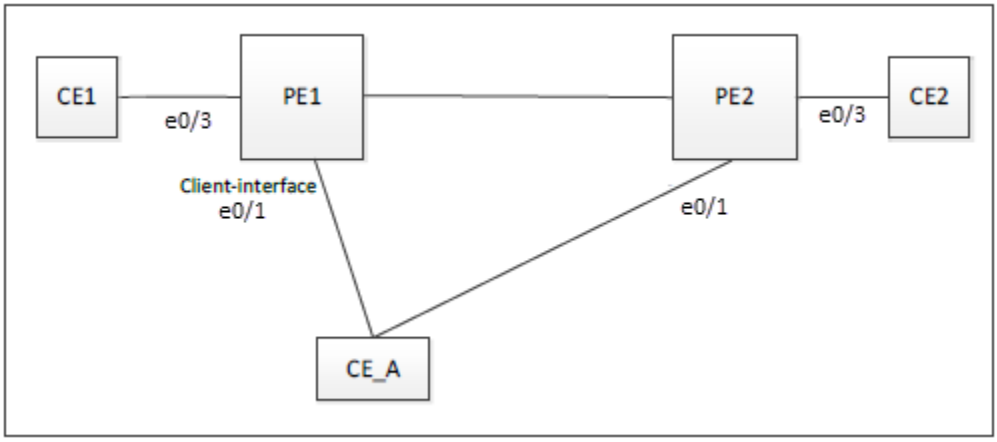
The following example displays all the MAC addresses learned from other VPLS PE nodes over MCT bridge domains.

```
device# show mac-address-table bridge-domain
Type Code - CCL:Cluster Client Local MAC
            CCR:Cluster Client Remote MAC
            CR:Cluster Remote MAC
BDId          Mac-address       Type        State      Ports/LIF/PW/T
501 (B)       0000.0500.0200    CR          Active     Tu 32769 (4.1.1.11)
501 (B)       0000.0500.0500    CCR         Active     Po 10.501
Total MAC addresses    :  2
```

Layer 3 routing over MCT

Layer 3 routing is supported for IPv4 and IPv6 BGP, OSPF, and IS-IS routing protocols on an MCT VLAN or bridge domain (BD). All local and remote Layer 3 devices appear logically on the same VLAN or BD.

The following diagram is the Layer 3 MCT data plane.



All devices are on the same VE and receive protocol Hello packets. Over the ICL link, the following packets are flooded:

- Hello and multicast packets from PE1 and PE2
- ARP and ND6 packets

Any two devices have direct Layer 3 communications.

- IP traffic is sent to the DA MAC of the target IP address or next-hop IP address.
- Traffic from CE_A to PE1 or CE1 might be sent through the PE2 link first, and switched by PE2 over the ICL link.

Configuration considerations

You must first create the VE interface for the MCT VLAN or bridge domain on the MCT pair.

Enabling L3 protocols is the same as enabling them on a VE interface.

For routes learned over Layer 3 protocols, the next-hop IP address is usually the peer IP address and not necessarily the MCT router address.

To bind the VE interface to an MCT VLAN or bridge domain, use the **router-interface ve** command under VLAN or bridge-domain configuration mode, respectively. The following example shows how to bind VE 200 to bridge domain 2.

```
device# configure terminal
device(config)# bridge-domain 2
device(config-bridge-domain-2)# router-interface ve 200
```

Note
Where supported, MPLS cannot be enabled for a VE over an MCT VLAN interface.

Layer 3 MCT VLAN configuration example

The following configuration example shows how to enable OSPFv2 and OSPFv3 protocols on PE1, PE2, and CE_A over VE 200 for the MCT member VLAN 2.

PE1:

```
router ospf
  area 0

ipv6 router ospf
  area 0

vlan 2
 router-interface Ve 200

interface Ve 200
 ipv6 address 2001::1/64
 ip address 10.2.2.1/24

 ip ospf area 0
 ipv6 ospf area 0
 !
 no shutdown
!
```

PE2:

```
router ospf
  area 0

ipv6 router ospf
  area 0

vlan 2
 router-interface Ve 200

interface Ve 200
 ipv6 address 2001::2/64
 ip address 10.2.2.2/24

 ip ospf area 0
 ipv6 ospf area 0
 !
 no shutdown
!
```

CE_A:

```
router ospf
  area 0

ipv6 router ospf
  area 0

vlan 2
 router-interface Ve 200

interface Ve 200
 ipv6 address 2001::10/64
 ip address 10.2.2.10/24

 ip ospf area 0
 ipv6 ospf area 0
```

```
 !
 no shutdown
!
```

Layer 3 MCT bridge-domain configuration example

The following configuration example shows how to enable OSPFv2 and OSPFv3 protocols on PE1, PE2, and CE_A over VE 200 for the MCT member bridge-domain 2.

PE1:

```
router ospf
  area 0

ipv6 router ospf
  area 0

interface Ve 200
 ipv6 address 2001::1/64
 ip address 10.2.2.1/24
!
 ip ospf area 0
 ipv6 ospf area 0
 !
 no shutdown

bridge-domain 2 p2mp
 router-interface Ve200
 logical-interface ethernet 0/1.2
 pw-profile default
 bpdu-drop-enable
 local-switching
!
```

PE2:

```
router ospf
  area 0

ipv6 router ospf
  area 0

interface Ve 200
 ipv6 address 2001::2/64
 ip address 10.2.2.2/24
!
 ip ospf area 0
 ipv6 ospf area 0
 !
 no shutdown

bridge-domain 2 p2mp
 router-interface Ve200
 logical-interface ethernet 0/1.2
 pw-profile default
 bpdu-drop-enable
 local-switching
!
```

CE_A:

```
router ospf
  area 0

ipv6 router ospf
  area 0

interface Ve 200
 ipv6 address 2001::10/64
 ip address 10.2.2.10/24
!
 ip ospf area 0
 ipv6 ospf area 0
 !
 no shutdown

bridge-domain 2 p2mp
 router-interface Ve200
 logical-interface ethernet 0/1.2
 pw-profile default
 bpdu-drop-enable
 local-switching

!
```

Using MCT with VRRP and VRRP-E

The MCT device that acts as the Virtual Routing Redundancy Protocol (VRRP) and VRRP Extended (VRRP-E) backup router performs as a Layer 2 switch to pass the packets to the VRRP or VRRP-E master router for forwarding. Through MAC synchronization, the VRRP or VRRP-E backup router learns the virtual MAC (VMAC) on the Inter-Chassis Link (ICL). The data traffic and control traffic both pass through the ICL cloud from the backup router. If VRRP-E short path forwarding is enabled, the backup router can forward the packets directly, instead of sending them to the master.

Note
Short path forwarding is only supported on VRRP-E.

In the MCT short path forwarding diagram below, when an ARP request from the S1 switch device is sent through the direct link to the VRRP or VRRP-E backup router (PE2), that same request is flooded through the ICL and received by the VRRP/E master router (PE1) for processing. When the ARP request is received by the PE1 device, PE1 sends a reply through the direct link to S1. If the ARP reply was received before the MAC address for the MCT on S1 is learned, the reply packet may be flooded to both the Customer Client Edge Port (CCEP) ports and ICL ports.

Using VRRP or VRRP-E, data traffic received from a client device on a backup router is Layer 2 switched to the master device. If VRRP-E short path forwarding is enabled, traffic received on the backup device may be forwarded by the backup if the route to the destination device is shorted than through the master device.

MCT short path forwarding configuration using VRRP-E example

In this example configuration, we are assuming that MCT is using the VRRP-E short path forwarding. When short path forwarding is enabled, packets from either the E1 or E2 devices with a destination of the E4 device can be routed through the PE 2 device which is a VRRP-E backup device. Short path forwarding is designed for load-balancing and allows packets to use the shortest path, and in this case, PE2 is directly connected to E4 so the packets will travel through PE2.

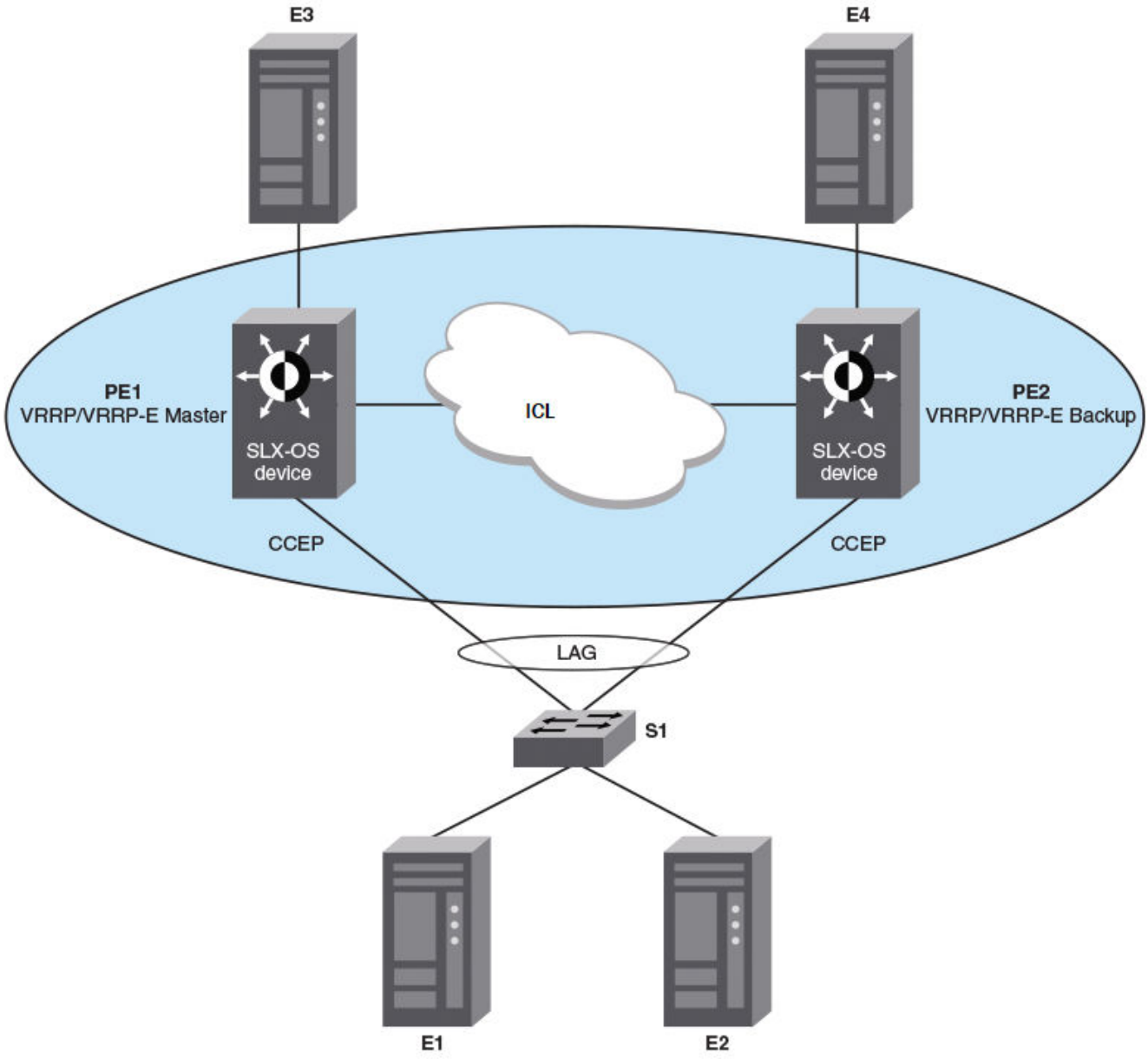


Figure 12: MCT short path forwarding

PE1 configuration

The following example configures the cluster for the PE1 router in the diagram. A VRRP-E priority value of 110 (higher than the device at PE2) allows the PE1 device to assume the role of VRRP-E master.

```
 interface Ethernet 0/3
  ip address 10.1.8.19/24
  no shutdown
!
 vlan 100
!
 interface Ethernet 0/5
  switchport
  switchport mode trunk-no-default-native
  switchport trunk allow vlan add 100
  no shutdown
!
 cluster <optional-cluster-name>
  peer 10.1.8.32
  peer-interface Ethernet 0/3
  peer-keepalive
    auto
!
 member vlan all
 member bridge-domain all
!
 vlan 100
  router-interface Ve 100
!
 protocol vrrp-extended
!
 interface Ve 100
 ip proxy-arp
 ip address 10.2.3.6/24
 vrrp-extended-group 1
   priority 110
   short-path-forwarding
   virtual-ip 10.2.3.4
 no shutdown
!
 interface Ve 100
 ipv6 address fe80::1:2 link-local
 ipv6 address 3313::2/64
 ipv6 vrrp-extended-group 1
 virtual-ip 3313::1
```

PE2 configuration

The following example configures the cluster for the PE2 router in the diagram. A VRRP-E priority value of 80 (lower than the device at PE1) allows the PE2 device to assume the role of a VRRP-E backup device.

```
interface Ethernet 0/3
ip address 10.1.8.32/24
no shutdown
!
vlan 100
!
interface Ethernet 0/7
switchport
```

```
switchport mode trunk-no-default-native
switchport trunk allow vlan add 100
no shutdown
!
cluster <optional-cluster-name>
peer 10.1.8.19
peer-interface Ethernet 0/3
peer-keepalive
  auto
!
member vlan all
member bridge-domain all
!
vlan 100
router-interface Ve 100
!
protocol vrrp-extended
!
interface Ve 100
ip proxy-arp
ip address 10.2.3.5/24
vrrp-extended-group 1
  priority 80
  short-path-forwarding
  virtual-ip 10.2.3.4
no shutdown
!
interface Ve 100
ipv6 address fe80::1:1 link-local
ipv6 address 3313::3/64
ipv6 vrrp-extended-group 1
virtual-ip 3313::1
```

MCT Use Cases

An L2 MCT solution can be deployed at the access, aggregation, and the core of the data center.

L2 MCT in the data center core

The following diagram shows a typical 3-tier data center where access and aggregation layers are running Layer 2 and the core is running Layer 2 and Layer 3. The access and aggregation can be standalone Extreme switches or any other third party switches.

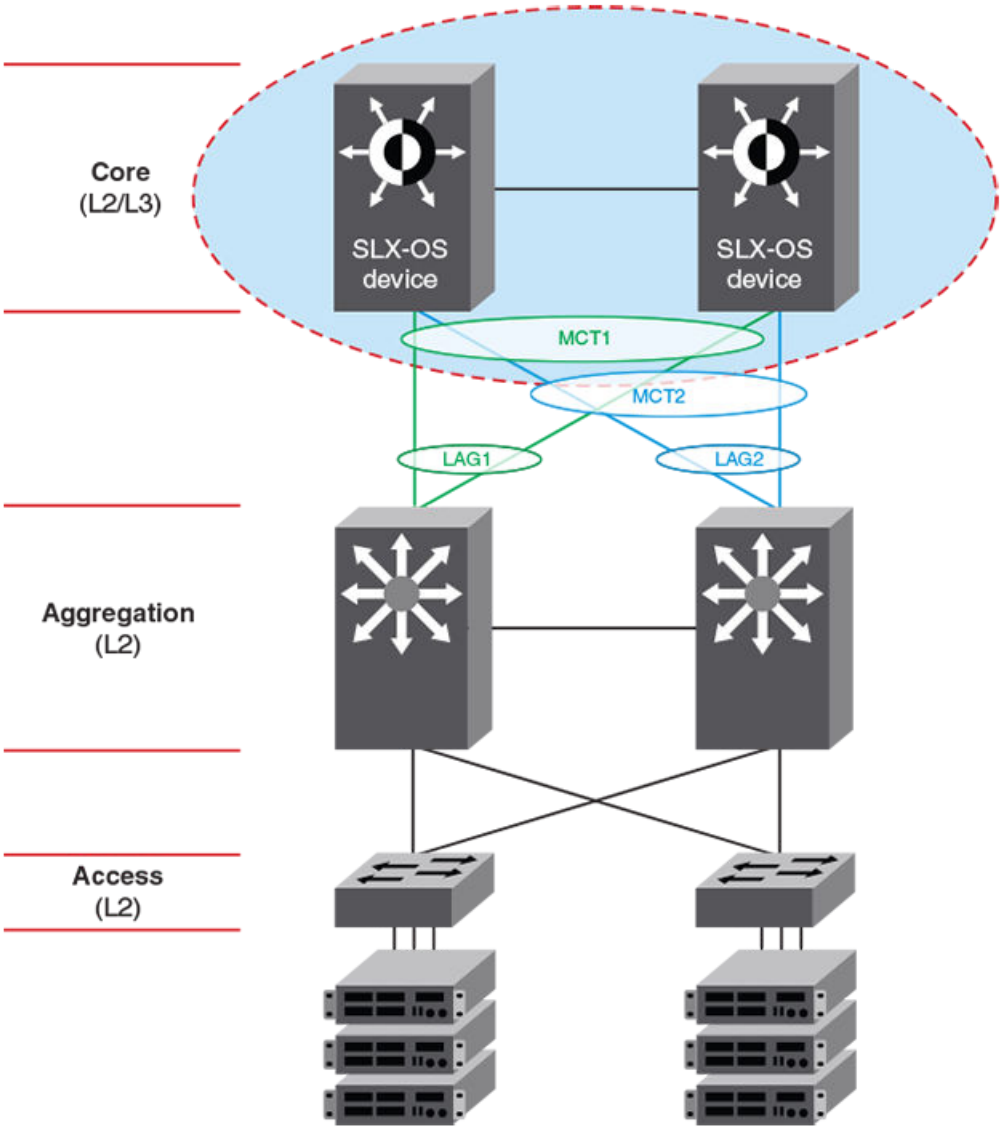


Figure 13: Typical 3-tier data center

Another variation of this use case is when the aggregation layer is a virtual cluster of switches which is transparent to SLX-OS devices in the core layer.

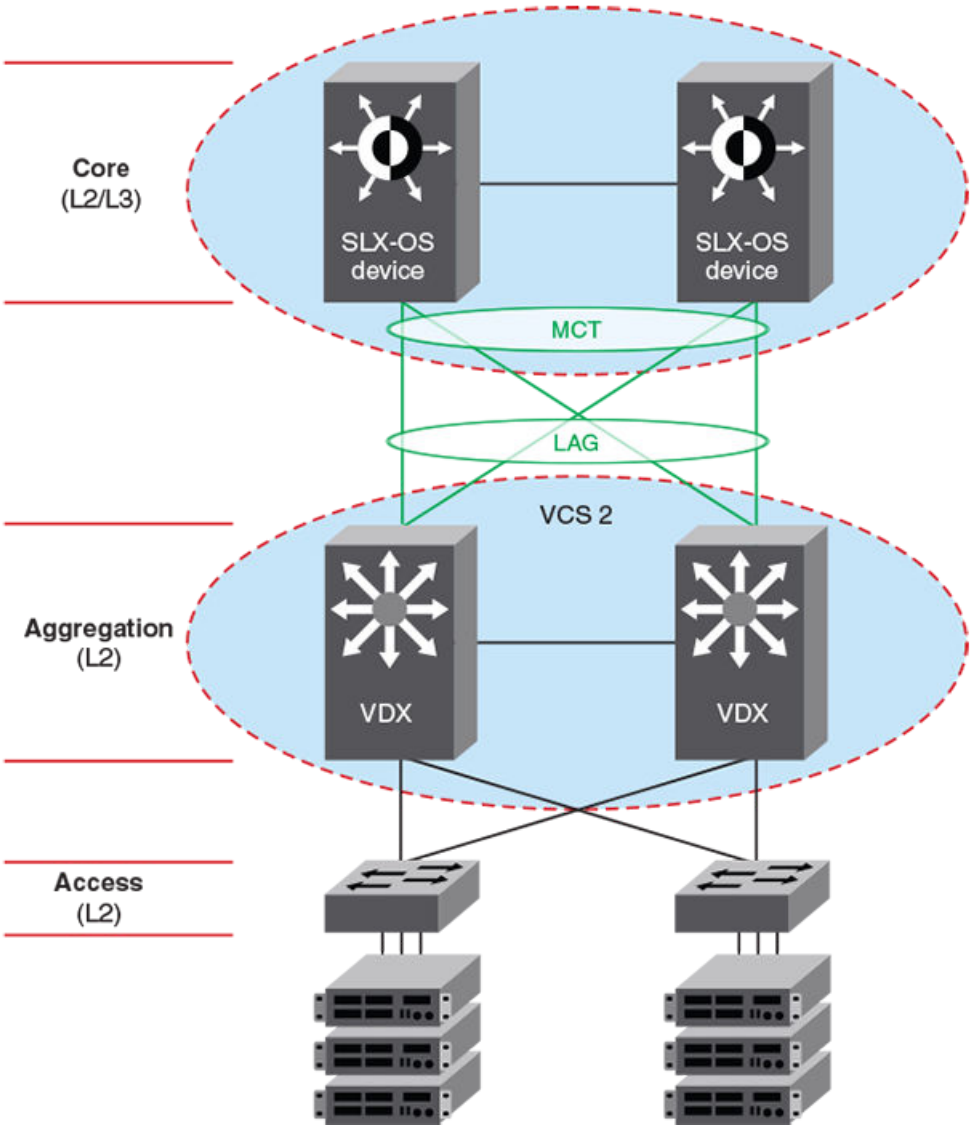


Figure 14: L2 MCT in the data center core connecting to a VCS

L2 MCT in a data center with a collapsed core and aggregation

The following diagram describes a scenario where VCS fabric of VDX 8870 switches is deployed at the access layer. With the availability of 10G and 40G interface, access switches can connect directly to the core without the need to have a separate aggregation layer.

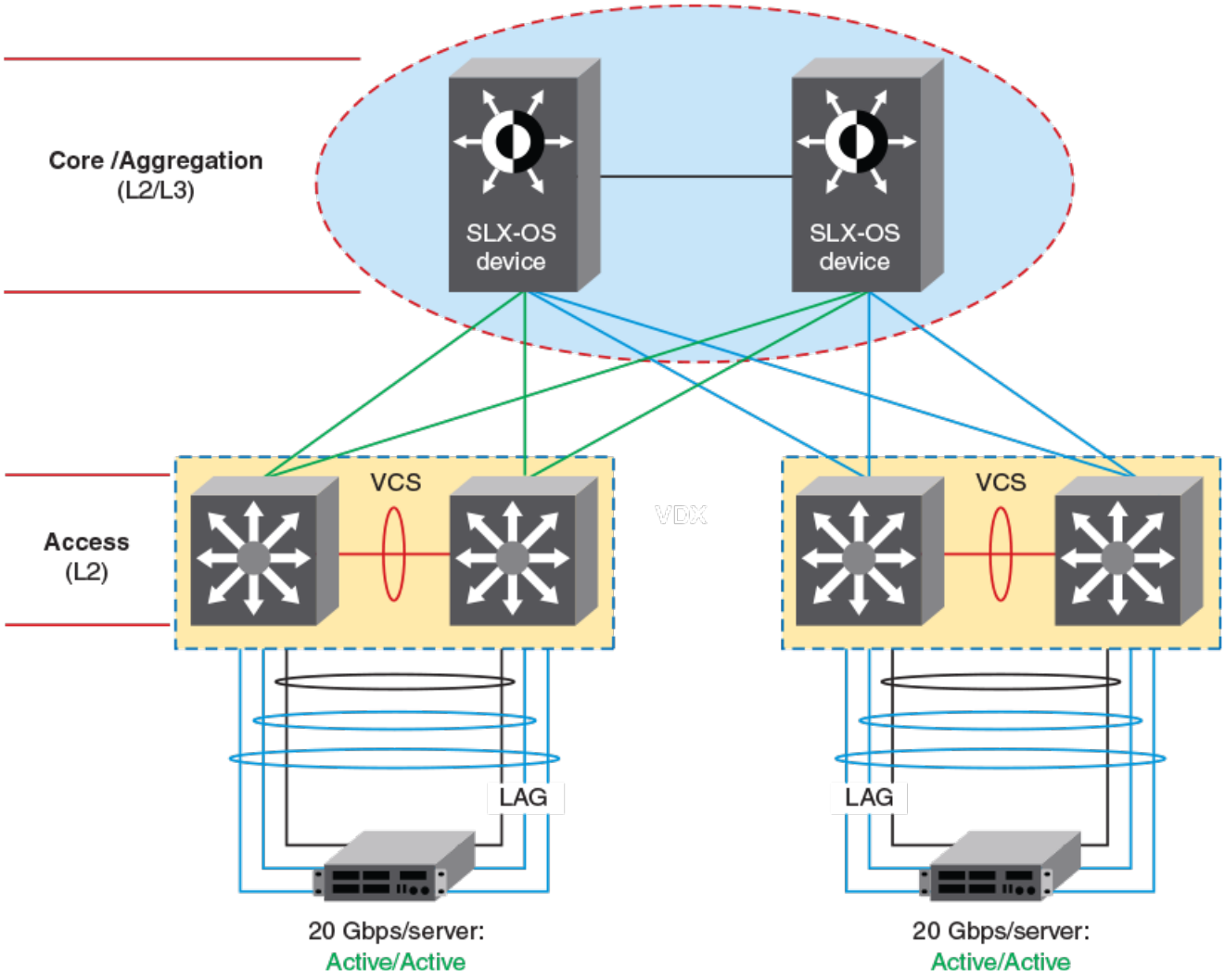


Figure 15: L2 MCT with collapsed core and aggregation

Logical Interfaces

Logical interfaces overview on page 135
Configuring a logical interface on a physical port or port-channel (LAG) on page 136

Logical interfaces overview

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

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

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

LIFs and bridge domains

A Layer 2 application for LIFs is for bridge domains (BDs). A BD is an infrastructure that supports the implementation of different switching technologies; it is essentially a generic broadcast/forwarding domain that is not tied to a specific transport technology. Bridge domains support a wide range of service endpoints, including regular Layer 2 endpoints and Layer 2 endpoints over Layer 3 technologies. Logical interfaces representing BD endpoints must be created before they can be bound to a BD. For more information and configuration details, refer to Bridge Domains on page 138.

Configuration considerations

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

- By default, when the LIF is created it is configured as "no shutdown."
- By default, when the LIF is created, it is "tagged" unless it is explicitly configured with the "untagged" option.
- Allowed LIF service instance ID ranges are from 1 through 12288.
- An LIF service instance ID has no correlation to the VLAN ID of the LIF.

- Each physical/LAG-based LIF must have an associated VLAN configured or else it will not be usable when the user attempts to add it to a service (such as VPLS, Layer 2). Such a configuration request to add the LIF to a service will be rejected.
- Once the LIF is associated with a Layer 2 service, its VLAN value cannot be changed or deleted unless it is first removed from the associated service. In case the LIF is not yet associated to a service, the user is free to remove the VLAN configuration or change the VLAN assignment.
- The **no** option to the **logical-interface** command can be applied at any time.
- The "untagged" configuration is allowed for only one LIF under the same physical port or LAG. If one LIF is already configured as untagged, all subsequent attempts on the same physical port or LAG are rejected.
- Once the "untagged" option is selected, it will only have one VLAN as the next classification option. There is no dual-tag support for the untagged case.
- In order to configure an untagged LIF, the main interface must be configured as "switchport mode trunk-no-default-native". If it is configured set to regular trunk mode, the native VLAN is already associated with a regular Layer 2 VLAN LIF and no explicit untagged LIF can be configured on that interface.
- Once the LIF is associated with a service (Layer 2) such as a bridge domain, its "untagged/tagged" configuration cannot be changed. The service instance or its current VLAN classification must be deleted by the user first and then added back with the proper "untagged/tagged" option.
- VLANs 4091 through 4095 are reserved VLANs and these should not be used as the VLAN ID for either the inner or outer VLAN of the LIF.
- The VLAN specified under the LIF ensures that such a VLAN is not already configured under the **switchport** command for a regular Layer 2 allowed VLAN.

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

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

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

1. Do the following to configure a logical interface on an Ethernet port.
   a. Enter global configuration mode.
   ```
   device# configure terminal
   ```
   b. Specify an Ethernet interface.
   ```
   device(config)# interface ethernet 0/6
   ```
   c. Enter the **switchport** command to configure the parent interface as switchport.
   ```
   device(conf-if-eth-0/6)# switchport
   ```
   d. Enter the **switchport mode trunk-no-default-native** command to enable an explicit untagged LIF to be configured.
   ```
   device(conf-if-eth-0/6)# switchport mode trunk-no-default-native
   ```

e. Enable the interface.

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

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

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

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

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

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

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

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

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

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

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

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

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

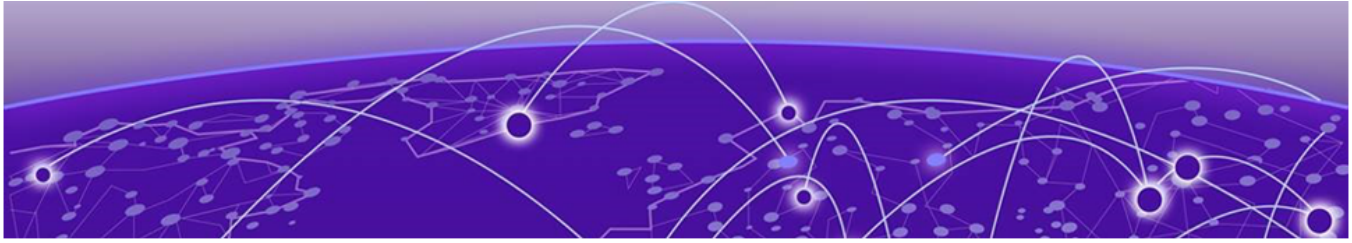
```
device(conf-if-eth-0/6)# logical-interface ethernet 0/6.120 name myLIF120

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

2. To configure a port-channel, configure the basic LIF parameters and options as in Step 1.

a. Specify a port-channel, set its mode to "trunk-no-default-native," and specify a logical interface service instance.

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

b. Repeat additional substeps in Step 1 as appropriate.



Bridge Domains

Bridge domain overview on page 138
Configuring a bridge domain on page 139
Displaying bridge-domain configuration information on page 140
Enabling statistics on a bridge domain on page 144
Displaying statistics for logical interfaces in bridge domains on page 144
Clearing statistics on bridge domains on page 145

Bridge domain overview

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

Bridge domains switch packets between a range of different endpoint types; for example, attachment circuit (AC) endpoints, Virtual Private LAN Service (VPLS) endpoints, Virtual Leased Line (VLL) endpoints, and tunnel endpoints.

The following are examples of bridge-domain capable services:

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

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

Bridge domain statistics

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

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

Note

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

Configuring a bridge domain

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

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

Perform the following task to configure a bridge domain.

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

```
device# configure terminal
```

2. Create a bridge domain.

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

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

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

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

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

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

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

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

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

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

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

   By default, local switching is enabled.

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

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

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

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

Displaying bridge-domain configuration information

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

```
device# show bridge-domain

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

Bridge-domain 1
-------------------------------
Bridge-domain Type: mp , VC-ID: 5
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx: 1207959555, Local switching: TRUE, bpdu-drop-enable:TRUE
PW-profile: 1, mac-limit: 128000
Number of Mac's learned:90000,     Static-mac count: 10,
VLAN: 100, Tagged ports: 2(2 up), Un-tagged ports: 0 (0 up)
Tagged ports: Eth 0/6, eth 0/8
Un-tagged ports:

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

Bridge-domain 2
```

```
-------------------------------
Bridge-domain Type: mp , VC-ID: 100
Number of configured end-points:  5 , Number of Active end-points: 4
VE if-indx:  NA, Local switching: FALSE, bpdu-drop-enable:FALSE
PW-profile: profile_1, mac-limit: 262144
Number of Mac's learned:90000,     Static-mac count: 10,
VLAN: 100, Tagged ports: 2(1 up), Un-tagged ports: 0 (0 up)
      Tagged ports: eth 0/10, eth 0/11
      Un-tagged ports:
VLAN: 150, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
      Tagged ports: eth 0/5
      Un-tagged ports:

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

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

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

```
device# show bridge-domain 501

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

```
Total VPLS peers: 1 (1 Operational):

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

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

```
show bridge-domain 501

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

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

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

```
device# show bridge-domain 501

Bridge-domain 501
------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
```

```
PW-profile: default, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth 0/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

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

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

```
device# show bridge-domain brief

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

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

- The following example shows how to display bridge-domain information for an Ethernet interface (0/2).

```
device# show bridge-domain interface ethernet 0/2
BDID 6, logical-interface eth0/2.100, VLAN 100,  tagged, UP
BDID 6, logical-interface eth0/2.101, VLAN 101,  tagged, UP
BDID 6, logical-interface eth0/2.102, VLAN 102,  tagged, UP
BDID 6, logical-interface eth0/2.1,   VLAN 4000, tagged, UP
BDID 7, logical-interface eth0/2.103, VLAN 103,  tagged, UP
BDID 7, logical-interface eth0/2.104, VLAN 104,  tagged, UP
```

- The following example shows how to display bridge-domain information for a port-channel interface (10).

```
device# show bridge-domain interface port-channel 10
BDID 6, logical-interface po10.200, VLAN 200, tagged, DOWN
BDID 6, logical-interface po10.201, VLAN 201, tagged, DOWN
BDID 7, logical-interface po10.202, VLAN 202, tagged, DOWN
BDID 7, logical-interface po10.203, VLAN 203, tagged, DOWN
BDID 7, logical-interface po10.204, VLAN 204, tagged, DOWN
```

Enabling statistics on a bridge domain

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

1. Enter the global configuration mode.

```
device# configure terminal
```

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

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

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

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

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

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

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

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

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

Displaying statistics for logical interfaces in bridge domains

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

```
device# show statistics bridge-domain

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

Bridge Domain 20 Statistics
Interface                    RxPkts          RxBytes          TxPkts
TxBytes
eth 1/6.400       821729           821729           95940360         95940360
eth 1/21.100        8884             8855               9684             9955
po  2.40          884484           885855           95969584         95484555
```

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

```
device# show statistics bridge-domain 1

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

Clearing statistics on bridge domains

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

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

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

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

VPLS and VLL Layer 2 VPN services

VPLS overview on page 146
Configuring a PW profile on page 159
Attaching a PW profile to a bridge domain on page 160
Configuring control word for a PW profile on page 160
Configuring PW control word on a bridge domain on page 161
Configuring flow label for a PW profile on page 162
Configuring PW flow label on a bridge domain on page 163
Configuring a static MAC address over an endpoint in a VPLS instance on page 164
Displaying MAC address information for VPLS bridge domains on page 165
Configuring a VPLS instance on page 165
Configuring a VLL instance on page 167
Routing VE over VPLS on page 168
Configuration example for VPLS with switching between ACs and network core on page 171
VPLS MAC withdrawal on page 172

VPLS overview

Note
VPLS and VLL Layer 2 VPN services are not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.

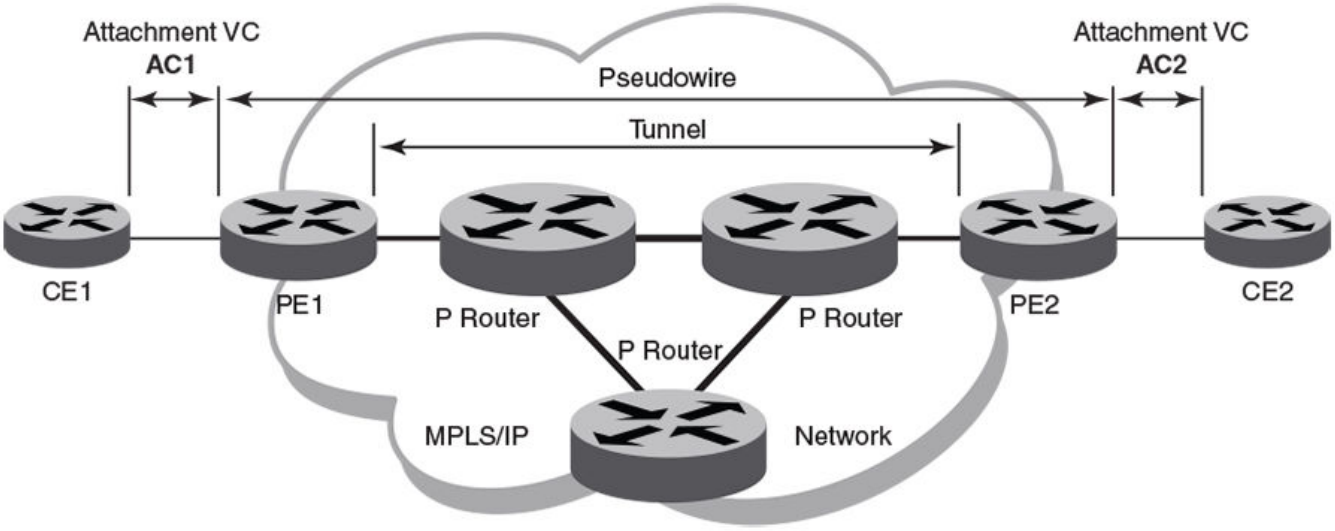
VPLS provides transparent LAN services across provider edge (PE) devices using Internet Protocol (IP) or Multiprotocol Label Switching (MPLS) as the transport technology.

Because it emulates LAN switching, VPLS is considered to be a L2 service that operates over Layer 3 (L3) clouds.

VPLS provides point-to-multipoint (p2mp) functionality while VLL is a special type of VPLS deployment that performs point-to-point (p2p) switching.

VLL is a special type of VPLS deployment that performs point-to-point switching.

The following figure shows a VPLS topology in which switched packets traverse a network.



AC1 and AC2 represent L2 connectivity between customer edge (CE) and provider edge (PE) devices.

Pseudowire is a circuit emulation infrastructure that extends L2 connectivity from CE1 to CE2 by way of PE1 and PE2. The tunnel is typically a L3 tunnel on which a L2 circuit is emulated.

In the case of a packet flowing from CE1 to CE2, the packet enters PE1 from CE1 after the forwarding database (FDB) is used to determine the destination MAC address. Then, a virtual connection (VC) label is imposed prior to encapsulation with the tunnel forwarding information, and the packet is sent out onto the wire towards the network core.

Figure 16: VPLS topology with switching between attachment circuits (ACs) and network core

Essentially, the topology in the preceding figure shows a L2 VPN enabling the transport of L2 traffic between two or more native Ethernet networks through an underlying Multiprotocol Label Switching (MPLS) provider network. Customer edge (CE) is the last mile and provider edge (PE) is the first mile node for packets transported towards the provider network. The provider intermediary network is an emulated switch (LAN) or wire (LINE) to the CE. The attachment circuit (AC) represents the logical link between the CE and PE.

An AC may be a port, IEEE 802.1q or IEEE 802.1ad (QinQ)) for Ethernet VPNs. A pseudowire (PW) or emulated wire is used as a transport mechanism to tunnel frames between PEs. A PW is characterized by a circuit identifier, which identifies the destination PE.

MPLS tunnels and paths are established by using routing protocols. PW circuits are established by using signaling.

The following figure shows a VPLS topology where switching occurs between two local AC endpoints. This implementation of VPLS does not use VC labels or a pseudowire.

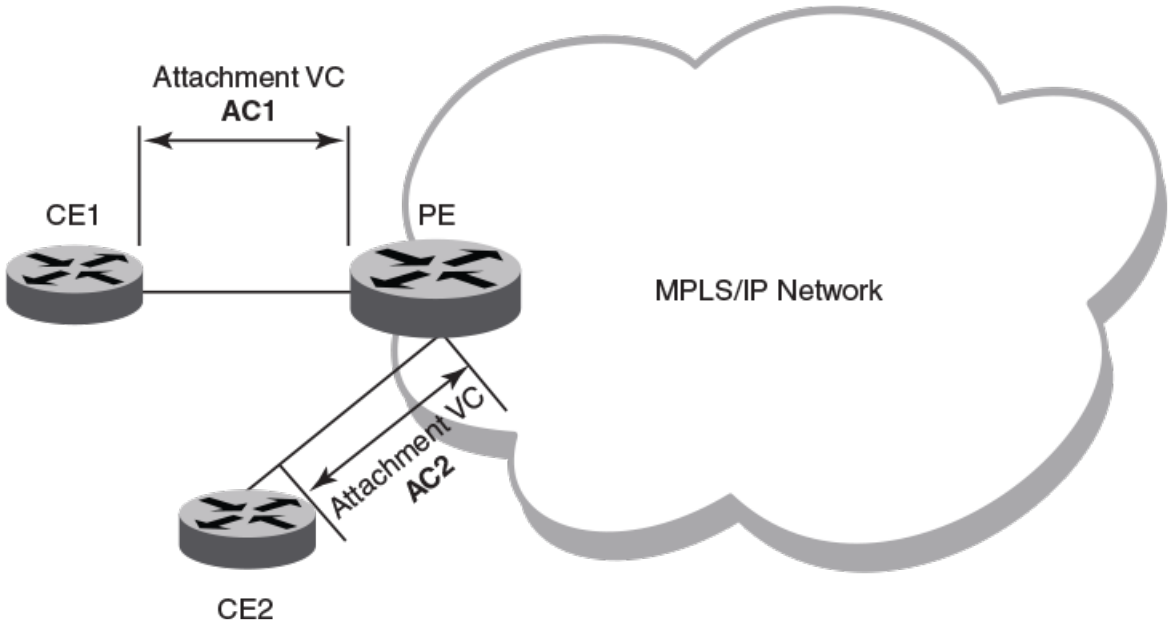


Figure 17: VPLS topology with local switching

The following figure shows a common VPLS deployment; an enterprise LAN service. The CE devices represent customer edge devices while the PE devices represent provider edge devices.

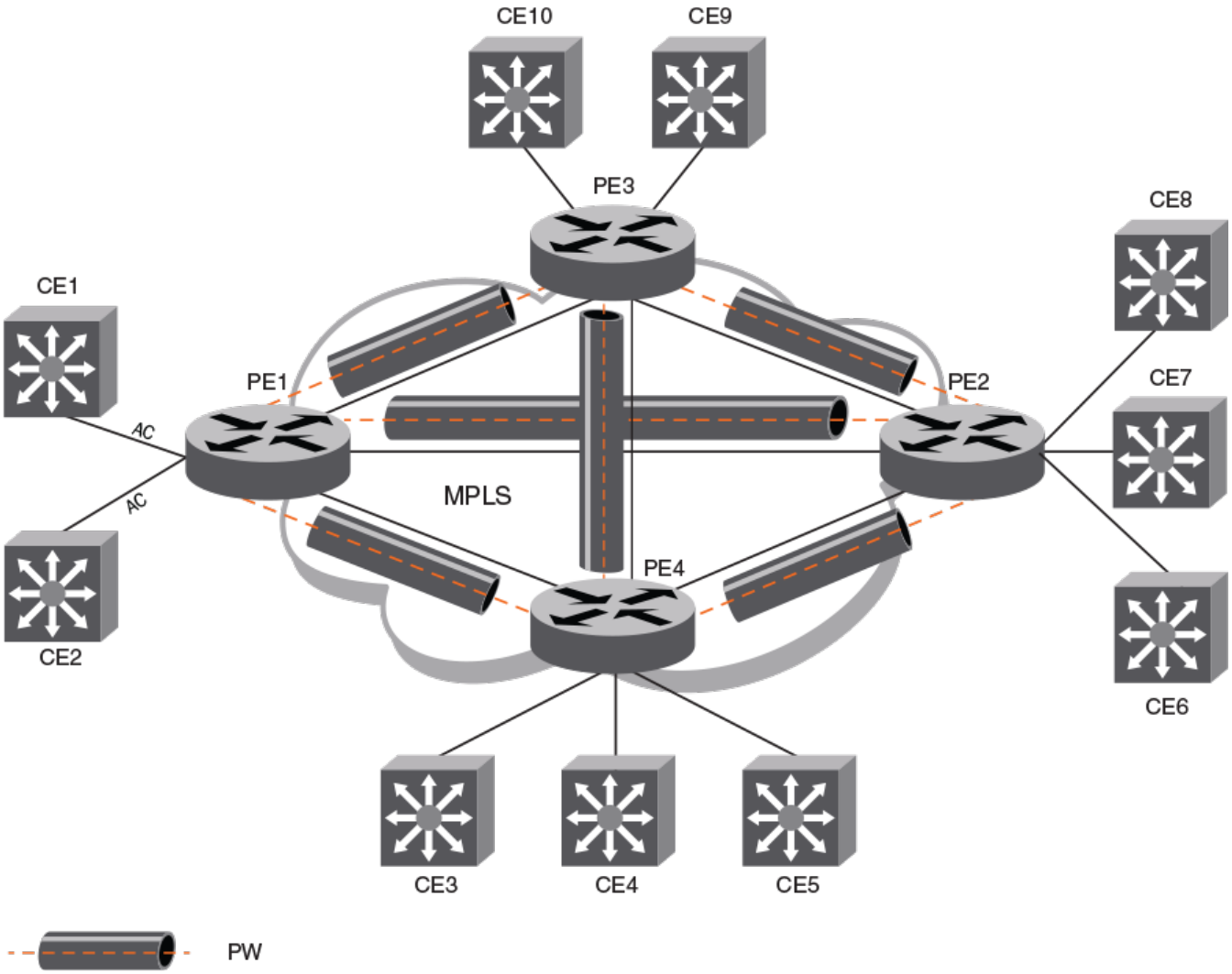


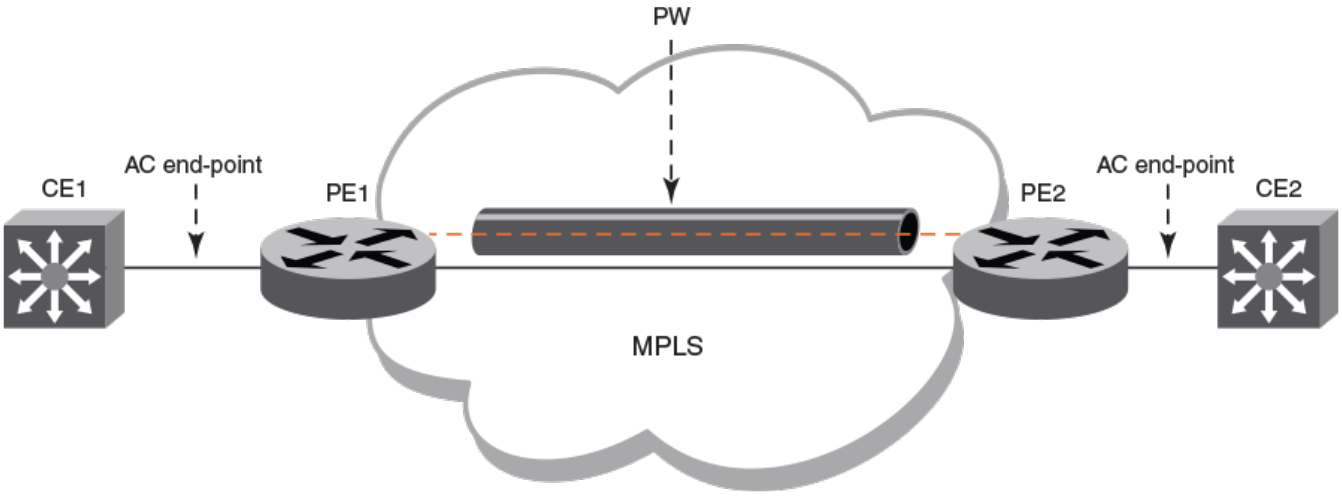
Figure 18: Enterprise LAN service (VPLS)

VLL

A VLL instance is a special type of VPLS deployment.

VLL provides point-to-point (p2p) connectivity between two access networks or endpoints. Typically, a VLL is used to connect two sites that are geographically apart.

The following figure depicts an enterprise VLL service.



CE1 and CE2 are the customer edge devices in geographically separate sites.

Pseudowire (PW) is a circuit emulation infrastructure that extends L2 connectivity from CE1 to CE2 by way of PE1 and PE2. The tunnel is typically a L3 tunnel on which a L2 circuit is emulated.

Figure 19: Enterprise leased line service (VLL)

VPLS service endpoints

Service endpoints can be categorized as:

- AC endpoints
- PW endpoints

An AC endpoint is a L2 link between a PE device and a CE device. The AC endpoint can be an untagged port, or a tagged port with one or more VLANs. AC endpoints with different VLAN tags can be configured in a single VPLS instance.

A VLL instance interconnects two AC endpoints through a pseudowire, while a VPLS instance forms a full mesh interconnection between multiple AC endpoints through multiple PWs.

The following endpoints are supported:

- port-vlan
- port-vlan-vlan
- PW

Both regular port and port-channel interfaces can be used to form port-vlan, untagged port-vlan, and port-vlan-vlan endpoints.

VPLS service endpoints are represented by logical interfaces (LIFs). By using LIFs, features that apply to regular interfaces, such as QoS, can be applied to VPLS service endpoints.

Local switching

When local switching is enabled, traffic is switched and flooded among AC endpoints in addition to between ACs and PWs. When local switching is disabled, the flooding between AC endpoints is suppressed.

When an unknown unicast packet is received on an AC endpoint and local switching is enabled, the packet is flooded to all other AC endpoints and PW endpoints in the VPLS instance. When local switching is disabled, the unknown packet is only flooded to the PW endpoints in the domain.

Regardless of the local switching configuration, an unknown unicast packet that is received on a PW endpoint is flooded to all AC endpoints.

By default, local switching is enabled.

In a VPLS instance that does not have a PW peer and where all endpoints are AC endpoints (Local VPLS), local switching must be enabled.

To avoid receipt of traffic with different VLAN tags on local endpoints, it is recommended that local switching is disabled in a bridge domain where the PW profile is configured with the VC mode option of **`raw-passthrough`**. Raw passthorugh mode is designed to forward packets between two VPLS peer devices and is not intended for use with local switching.

Pseudowires

An Ethernet pseudowire is logically viewed as an L2 nexthop (VC label) that is reachable through an L3 nexthop (LDP label).

The frames from an AC endpoint packet are sent through an ingress pseudowire interface (which abstracts the transport path and packet encapsulations) towards the remote PE. An egress pseudowire interface then abstracts the packet received from a remote PE and hands it over to the corresponding AC end-point.

A pseudowire interface is unidirectional.

PWs support the following underlying MPLS tunnels:

- LDP – Single Path LSP
- RSVP – Single Path LSP
- RSVP – Pri/Sec (Act LSP)
- RSVP – Pri/Sec (Pas LSP)
- FRR: Adaptive LSP (Make Before Break)
- FRR: Protected & detour (1:1)

PWs do not support the following underlying MPLS tunnels:

- FRR: Protected & Bypass (N:1)
- LDP – Multipath LSP (ECMP)
- LDP over RSVP

Pseudowire operation

A pseudowire is operational when the following conditions are met:

- VC signaling is established.
- The L3 reachability of the PW peer is resolved.
- At least one AC endpoint within the bridge domain is up.

A pseudowire is non-operational when the following conditions are met:

- No logical interface is configured for the VPLS instance.
- All AC endpoints are non-operational.

Supported pseudowire features

- LSP Load Balancing—Load balancing across a maximum of 16 underlying MPLS tunnels.
- Assigned LSP—A maximum of 32 LSPs can be assigned.
- Specific COS—The underlying MPLS tunnel with the closest CoS value is selected for the transport
- Raw, raw-passthrough, or tagged mode—Can be configured by way of the PW profile that is associated with the bridge domain.
- MTU and MTU check—Can be configured by way of the PW profile that is associated with the bridge domain.
- Uniform and pipe mode for QoS
- Statistics—Egress and ingress statistics are supported but must be enabled in the bridge-domain configuration by using the **`statistics`** command

Unsupported pseudowire features

- Auto-discovery of peers
- PW redundancy
- Static PW peers
- VC MAC withdraw
- Status TLV update
- VEoVPLS
- OAM
- Multicast snooping
- Extended counters
- High availability—Process restart
- High availability—ISSU

VLAN tag manipulation (vc-mode) on pseudowires

The following table describes VC modes that are supported on PWs.

Table 34: VC modes supported on pseudowires

VC mode	Description
Raw	At VC label imposition, when a tagged packet is received on a tagged AC endpoint, the VLAN tag is removed before it is sent out on the wire. When an untagged packet is received on an untagged AC endpoint it is encapsulated as is and sent out on the wire.
Raw-passthrough	Enables interoperation with third-party devices. When a packet that is destined for a remote peer is received on either a tagged or untagged AC endpoint, it is encapsulated in an MPLS header and sent on to the MPLS cloud without adding or removing VLAN tags. When a packet that is destined for a local endpoint is received on either a tagged or untagged AC endpoint, the MPLS header is removed before sending it on to the local endpoint; VLAN tags in the original packet are not changed in any way.
Tagged	At VC label imposition, when a tagged packet is received on a tagged AC endpoint, the packet is encapsulated as is and sent out on the wire. When a packet is from a dual-tagged AC endpoint, the outer VLAN tag is removed and only the inner VLAN TAG is sent out on the wire. When an untagged packet is received on an untagged AC endpoint, a dummy tag is added and it is sent out on the wire.

The VC mode is agreed by PE peer devices during the pseudowire signaling process.

A single VPLS instance can have a mixture of tagged and untagged endpoints.

When the VC mode is changed on a device, the PWs are torn down are re-established except in the cases of a change from raw to raw-passthrough or a change from raw-passthrough to raw. The traffic impact is minimal (because PWs are not torn down and re-established) when the VC mode is changed from raw to raw-passthrough (or vice versa).

VC mode is configured by specifying the **vc-mode** option for the **pw-profile** command.

Note
When a pseudowire profile is attached to a bridge domain, on which routing is enabled (by using the **router-interface** command), you are not allowed to change the pseudowire profile **vc-mode** configuration to **raw**.

PW statistics

PW statistics are enabled or disabled per bridge domain and apply to all the PWs that are part of the bridge domain. The logical interfaces for inbound and outbound statistics are shared resources. Hence, the corresponding PW is operational only when these hardware resources are available.

PW statistics are configured using the **statistics** command in bridge domain configuration mode. For further information about enabling statistics on a bridge domain, refer to *Bridge Domains*.

Pseudowire control word and flow label

- PW control word supports optimal transport of Ethernet Protocol Data Units (PDUs) over an MPLS PSN.
- PW flow label supports improved load balancing of PW traffic over an MPLS PSN.

PW control word

To ensure that a PW operates correctly over an MPLS PSN, the PW traffic is normally transported over a single network path, even when equal-cost multiple-paths (ECMPs) exist between the ingress and egress PW provider edge (PE) devices. Transporting PW traffic over a single network prevents misordered packet delivery to the egress PE device.

Because the standard MPLS label stack does not contain an explicit protocol identifier, label switching routers (LSRs) in ECMP implementations may examine the first nibble after the MPLS label stack to determine if a labeled packet is an IP packet. When the source MAC address of an Ethernet frame that is carried over a PW (without a control word present) begins with 0x4 or 0x6, it could be mistaken for an IPv4 or IPv6 packet. Depending on the configuration and topology of the MPLS network, this misinterpretation could lead to a situation in which all packets for a specific PW do not follow the same path and may increase out-of-order frames on the specific PW, or cause packets to follow a different path than actual traffic.

PW control word (RFC 4385) enables all packets for a specific PW to follow the same path over an MPLS PSN.

The following figure shows the format of the PW control word.

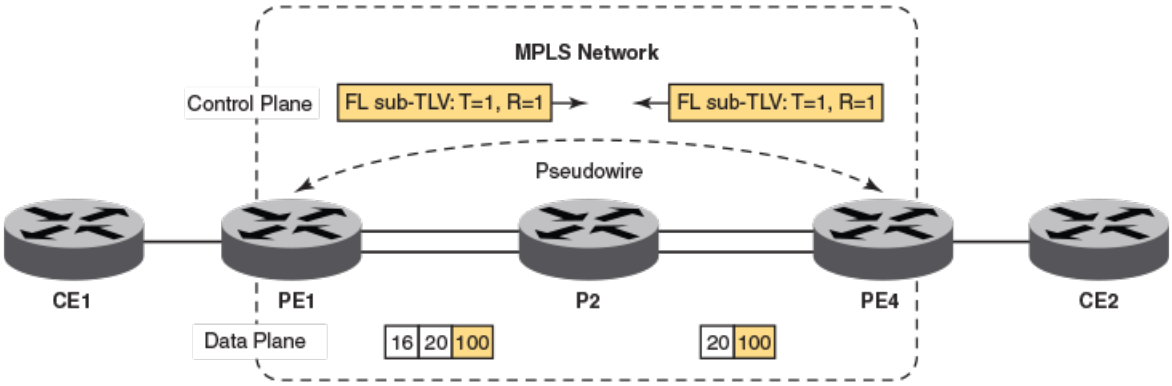
The first 4 bits are set to 0 to indicate PW data and distinguish a PW packet from an IP packet. The next 12 bits are reserved.

The last 16 bits carry a PW-specific sequence number that guarantees ordered frame delivery. A sequence number value of 0 indicates that the sequence number check algorithm is not used.

Figure 20: PW control word format

Note
PW control word is a nonconfigurable, global system value.

The following figure shows the operation of PW control word in a PW over MPLS network configuration.



- The ingress device (PE1) initiates the PW setup, including the capability to transmit and receive PW control word by using the new interface parameter TLV.
- The egress device (PE4) signals a control word-handling capability.
- When PE1 receives traffic from the customer edge device (CE1), it pushes the label stack onto the traffic control word and forwards the packet to P2.
- P2 uses the field immediately after the bottom of the MPLS label stack to identify the payload as non-IP, IPv4 or IPv6 and forwards the packet to PE4.
- PE4 receives the packet with two labels; the PW label, which identifies the PW, and PW control word, which is discarded without processing.

Figure 21: PW over MPLS

Pseudowire (PW) control word is configured by enabling control word for either a PW profile or for PW-related devices in a bridge domain. To enable control word for a PW profile, use the **control-word** command in PW profile configuration mode. To enable control word for PW-related devices in a bridge domain, use the **peer** command in bridge domain configuration mode.

PW flow label

Some PWs transport high volumes of traffic that comprise multiple separate traffic flows (for example, all packets for the same source-destination pair for a Transport Control Protocol [TCP] connection is a specific traffic flow).

To avoid latency and jitter, it is important that packets for a specific traffic flow are mapped to the same links along the path to the egress device.

PWs that carry multiple traffic flows require only the preservation of packet ordering in the context of an individual traffic flow and do not require the preservation of packet order for all traffic carried on the PW.

In normal cases, routers use source and destination IP addresses, protocol type, and TCP or UDP port numbers as keys in a load-balancing algorithm to find the appropriate outgoing interface. In MPLS networks, deep packet inspection may be needed for LSRs to identify such keys.

PW flow label is a mechanism that supports load balancing in an MPLS network, without the need for deep packet inspection by LSRs.

The PW flow label is added to the bottom of the label stack, by the ingress LSR (which has complete information about the packet) before it pushes the label stack. Transit LSRs can use the entire label stack (including the flow label) as a key for a load-balancing algorithm.

The PW flow label allows PWs to leverage the availability of, for example, equal-cost multiple-path (ECMP) between LSRs in an MPLS PSN. The ingress PE device maps individual traffic flows within a PW to the same flow label. Label Distribution Protocol (LDP) uses the PW flow label to map packets for a specific traffic flow to the same links along the path and to ensure that packets for the specific traffic flow follow the same path over the MPLS PSN. The PW flow label is discarded at the egress PE device.

Note
PW flow label load balancing is not supported when the incoming packet contains more than three labels.

The following figure shows packet encapsulation when flow label is enabled for PW traffic.

Normal Packet Encapsulation for PW traffic

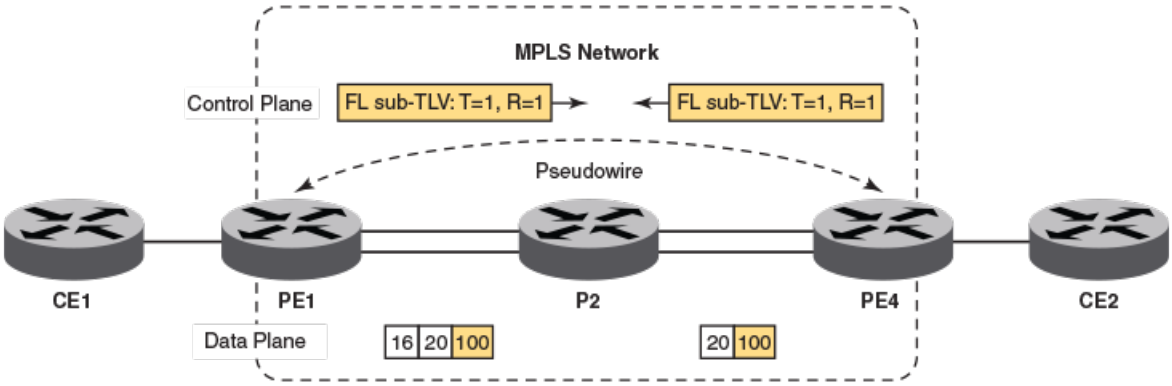
Packet Encapsulation with Flow Label for PW traffic

PW flow label is not a reserved value (that is, in the range from 0 through 15). To prevent any processing of flow label at the egress LSR, the TTL value of the flow label is set to 0.

Note
PW flow label is not signaled between PE routers. The flow label is generated locally and pushed to the bottom of the stack by the ingress LSR.

Figure 22: PW flow label

The following figure, which shows a PW over MPLS network configuration, describes the operation of PW flow label.



- The ingress device (PE1) initiates the PW setup, including the capability to transmit and receive PW flow label by using the new interface parameter TLV.
- The egress device (PE4) signals a flow label-handling capability.
- When PE1 receives traffic from the customer edge router, it uses a hash algorithm based on the keys (destination MAC address, source MAC address, and so on) to generate a flow label. It then pushes the label stack onto the flow label (S=1, TTL=0) and forwards the packet to P2.
- The P2 router uses the bottom label from the stack to ensure that a particular traffic flow (packets for a specific source-destination pair) follows the same path through transit routers.
- PE4 receives the packet with two labels with the top label being the PW label, which is used to identify the pseudowire, and the flow label, which is discarded without processing.

Note
When load balancing is based only on PW flow label, the transit device (P2) must be configured to include the bottom-of-stack (BOS) label by using the **`lag hash bos start`** command.

Figure 23: PW over MPLS

Pseudowire (PW) flow label is configured either by enabling flow label either for a PW profile or the bridge domain to which the PW is attached.

Pseudowire (PW) flow label is configured by enabling flow label for either a PW profile or PW-related devices in a bridge domain. To enable flow label for a PW profile, use the **`flow-label`** command in PW profile configuration mode. To enable flow label for PW-related devices in a bridge domain, use the **`peer`** command in bridge domain configuration mode.

Supported VPLS features

- Local VPLS (without PWs)
- VPLS—untagged, single-tagged, and dual-tagged endpoints
- Flooding of L2 BPDUs in VPLS
- VPLS tagged, raw, and raw-passthrough modes for virtual circuits

- Dynamic LAG support for VPLS endpoints
- VPLS MTU enforcement
- VPLS static MAC address support for AC endpoints
- VPLS over Multi-Chassis Trunking (MCT)
- Virtual Ethernet (VE) over VPLS
- Layer 3 ACLs. For details, refer to the ACLs > "Applying Layer 3 ACLs to interfaces" section of the *Extreme SLX-OS Security Configuration Guide*.

Configuration of VPLS and VLL

To configure a VPLS or VLL instance, you must complete the following tasks:

- Configure a bridge domain.
- Configure a VC identifier.
- Configure logical interfaces for AC endpoints.
- (Optional) Configure a pseudowire (PW) profile.
- Configure peer IP addresses. Configuring peer IP addresses creates PW endpoints.

Note
VPLS (or VLL) configuration is separate from the underlying IP or MPLS configuration. MPLS tunnels need to be brought up separately. For further information about the configuration of MPLS tunnels, refer to the *Extreme SLX-OS MPLS Configuration Guide*.

QoS treatment in VPLS packet flow

On the ingress label-edge router (LER), the final EXP value for the VC label is not dependant on the CoS value in the VC-peer configuration.

By default, for traffic flowing from a CE device to a PE device, 3 bits of the PCP field from the incoming Ethernet frame header are extracted and mapped to an internal CoS value by way of an ingress CoS map. This internal value is then mapped to an outgoing CoS value by way of an egress CoS map. The outgoing CoS value is then inserted into the EXP field in the outgoing VC label. When incoming traffic does not have VLAN tag, the default PCP value that is configured on a port is used.

In the case of traffic received from the network core side, by default the EXP field from the incoming VC label is mapped to an internal CoS value by way of an ingress CoS map. This internal value is then mapped to an outgoing CoS value by way of an egress CoS map. The outgoing CoS value is then inserted into the PCP field in the Ethernet frame header going out to the CE device.

On the egress LER, the CoS value for the VC-peer configuration is not dependant on the final EXP value for the VC label.

The following table shows ingress and egress behavior for different global, tunnel and PW configuration combinations.

Table 35: Ingress and Egress LER behavior

Global	Tunnel	PW	AC to PW (per path)	PW to AC (per PW)
Uniform	No CoS	No CoS	Uniform	Uniform
Uniform	CoS	No CoS	Pipe	Uniform
Uniform	No CoS	CoS	Uniform	Pipe
Uniform	CoS	CoS	Pipe	Pipe
Pipe	No CoS	No CoS	Pipe	Pipe
Pipe	CoS	No CoS	Pipe	Pipe
Pipe	No CoS	CoS	Pipe	Pipe
Pipe	CoS	CoS	Pipe	Pipe

Configuring a PW profile

A pseudowire profile can be shared across multiple bridge domains. Complete the following task to configure a PW profile.

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

```
device# configure terminal
```

2. Create a PW profile and enter configuration mode for the profile.

```
device(config)# pw-profile pw_example
```

3. Configure the virtual connection mode for the profile.

```
device(config-pw-pw_example)# vc-mode tag
```

   In this example, tag mode is configured for the PW profile, pw_example.

4. Configure a maximum transmission unit (MTU) of 1300 for the PW profile.

```
device(config-pw-pw_example)# mtu 1300
```

5. Enforce an MTU check during PW signaling.

```
device(config-pw-pw_example)# mtu-enforce true
```

The following example creates a PW profile named pw_example and configures attributes for the profile.

```
device# configure terminal
device(config)# pw-profile pw_example
device(config-pw-pw_example)# vc-mode tag
device(config-pw-pw_example)# mtu 1300
device(config-pw-pw_example)# mtu-enforce true
```

Attaching a PW profile to a bridge domain

Before it is attached to a bridge domain, the PW profile must be configured. Perform the following task to attach a PW profile to a bridge domain.

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

```
device# configure terminal
```

2. Enter bridge domain configuration mode.

```
device(config)# bridge-domain 501
```

3. Configure a PW profile for the bridge domain.

```
device(config-bridge-domain-501)# pw-profile pw_example
```

4. Return to privileged EXEC mode.

```
device(config-bridge-domain-501)# end
```

5. Verify the configuration.

```
device# show bridge-domain 501

Bridge-domain 501
-------------------------------
Bridge-domain Type: MP , VC-ID: 501
Number of configured end-points:  2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
PW-profile: pw_example, mac-limit: 0
VLAN: 501, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth1/6.501
Un-tagged Ports:
Total VPLS peers: 1 (1 Operational):

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

The following example shows how to attach a PW profile named pw_example to bridge domain 501.

```
device# configure terminal
device(config)# bridge-domain 501
device(config-bridge-domain-501)# pw-profile pw_example
```

Configuring control word for a PW profile

Before completing the following task, the PW profile on which control word is to be enabled must be configured. An example is provided at the end of this task that shows all the steps in order.

PW control word configuration enables all packets for a specific PW to follow the same path over the MPLS network supporting, for example, the optimal transport of Ethernet Protocol Data Units (PDUs) over the network.

Perform the following task to configure PW control word for a PW profile.

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

```
device# configure terminal
```

2. Enter PW profile configuration mode

```
device(config)# pw-profile pw_example
```

3. Enable control word for the PW profile.

```
device(config-pw-profile-pw_example)# control-word
```

Note
When control word is enabled for a previously configured PW, control word capabilities between PE devices are activated only after LDP neighbors are cleared by using the **clear mpls ldp neighbor** command. For further information on clearing LDP neighbors, refer to *Extreme SLX-OS MPLS Configuration Guide*.

The following example shows how to configure a PW profile and enable control word for the profile. It then shows how to attach the PW profile to a bridge domain.

```
device# device# configure terminal
device(config)# pw-profile pw_example
device(config-pw-pw_example)# vc-mode tag
device(config-pw-pw_example)# mtu 1300
device(config-pw-pw_example)# mtu-enforce true
device(config-pw-pw_example)# control-word
device(config-pw-pw_example)# exit
!
device(config)# bridge-domain 502
device(config-bridge-domain-502)# pw-profile pw_example
```

Configuring PW control word on a bridge domain

Before completing the following task, the relevant bridge domain must be configured. An example is provided at the end of this task that shows all the steps in order.

PW control word configuration enables all packets for a specific PW to follow the same path over the MPLS network supporting, for example, the optimal transport of Ethernet Protocol Data Units (PDUs) over the network.

Perform the following task to configure PW control word for a peer device in a bridge domain.

1. Enable control word for the PW profile.

```
device# configure terminal
```

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

```
device# configure terminal
```

   b. Enter bridge domain configuration mode. The following example shows how to enter configuration mode for bridge domain 502.

```
device(config)# bridge-domain 502
```

c. Enable control word for a PW-related peer device (10.9.9.9) that is configured on the bridge domain.

```
device(config-bridge-domain-502)# peer 10.9.9.9 control-word
```

2. Return to privileged EXEC mode.

```
device(config-pw-profile-pw_example)# end
```

3. Verify the configuration.

```
show bridge-domain 502

Bridge-domain 502
-----------------------------
Bridge-domain Type: MP, VC-ID: 502 MCT Enabled: FALSE
Number of configured end-points:  3, Number of Active end-points: 1
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
MAC Withdrawal: Disabled
PW-profile: default, mac-limit: 0
VLAN: 502, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth0/5.502
Un-tagged Ports:
Total VPLS peers: 2 (0 Operational):
VC id: 502, Peer address: 10.9.9.9, State: Wait for LSP tunnel to Peer, uptime: -
Load-balance: True, Cos Enabled: False,
Tunnel cnt: 0
Assigned LSPs count:2 Assigned LSPs:q502 q752
Local VC lbl: 0, Remote VC lbl: 0,
Local VC MTU: 1500, Remote VC MTU: 0,
Local VC-Type: 5, Remote VC-Type: 0
Local PW preferential Status: Active, Remote PW preferential Status: Active
Local Flow Label Tx: Enabled, Local Flow Label Rx: Enabled
Remote Flow Label Tx: Enabled, Remote Flow Label Rx: Enabled
Local Control Word: Enabled, Remote Control Word: Enabled
```

The following example shows how to configure a bridge domain on which control word is enabled for a PW-related peer.

```
device(config)# bridge-domain 502
device(config-bridge-domain-502)# vc-id 8
device(config-bridge-domain-502)# logical-interface ethernet 0/5.15
device(config-bridge-domain-502)# logical-interface port-channel 2.200
device(config-bridge-domain-502)# peer 10.9.9.9 load-balance control-word
device(config-bridge-domain-502)# peer 10.12.12.12 control-word
device(config-bridge-domain-502)# peer 10.12.12.12 lsp lsp1 lsp2
device(config-bridge-domain-502)# local-switching
device(config-bridge-domain-502)# bpdu-drop-enable
device(config-bridge-domain-502)#
device(config-pw-pw_example)# exit
```

Configuring flow label for a PW profile

Before completing the following task, the PW profile on which flow label is to be enabled must be configured. An example is provided at the end of this task that shows all the steps in order.

Flow label configuration improves load balancing of PW traffic over an MPLS network, particularly in the context of PWs that transport high volumes of traffic that are comprised of multiple individual traffic flows (for example, the same source-destination pair for a Transport Control Protocol (TCP) connection is an individual traffic flow).

Perform the following task to configure flow label for a PW profile.

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

```
device# configure terminal
```

2. Enter PW profile configuration mode.

```
device(config)# pw-profile pw_example
```

3. Enable flow label for the PW profile.

```
device(config-pw-profile-pw_example)# flow-label
```

The following example shows how to configure a PW profile and enable flow label on the profile. It then shows how to attach the PW profile to a bridge domain.

```
device# device# configure terminal
device(config)# pw-profile pw_example
device(config-pw-pw_example)# vc-mode tag
device(config-pw-pw_example)# mtu 1300
device(config-pw-pw_example)# mtu-enforce true
device(config-pw-pw_example)# flow-label
device(config-pw-pw_example)# exit

device(config)# bridge-domain 5
device(config-bridge-domain-5)# pw-profile pw_example
```

Configuring PW flow label on a bridge domain

Prior to completing the following task, the bridge domain on which flow label is to be enabled must be configured. There is an example at the end of this task that shows all the steps in order.

Flow label configuration improves load balancing of PW traffic over an MPLS network, particularly in the context of PWs that transport high volumes of traffic that are comprised of multiple individual traffic flows (for example, the same source-destination pair for a Transport Control Protocol (TCP) connection is an individual traffic flow).

Perform the following task to configure PW flow label on a bridge domain.

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

```
device# configure terminal
```

2. Enter bridge domain configuration mode. The following example shows how to enter configuration mode for bridge domain 502.

```
device(config)# bridge-domain 502
```

3. Enable flow label for the PW-related peer devices that are configured on the bridge domain.

```
device(config-bridge-domain-502)# peer 10.9.9.9 flow-label
```

4. Verify the configuration.

```
show bridge-domain 502

Bridge-domain 502
-------------------------------
Bridge-domain Type: MP, VC-ID: 502 MCT Enabled: FALSE
Number of configured end-points:  3, Number of Active end-points: 1
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
MAC Withdrawal: Disabled
```

```
PW-profile: default, mac-limit: 0
VLAN: 502, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth0/5.502
Un-tagged Ports:
Total VPLS peers: 2 (0 Operational):
VC id: 502, Peer address: 10.9.9.9, State: Wait for LSP tunnel to Peer, uptime: -
Load-balance: True, Cos Enabled: False,
Tunnel cnt: 0
Assigned LSPs count:2 Assigned LSPs:q502 q752
Local VC lbl: 0, Remote VC lbl: 0,
Local VC MTU: 1500, Remote VC MTU: 0,
Local VC-Type: 5, Remote VC-Type: 0
Local PW preferential Status: Active, Remote PW preferential Status: Active
Local Flow Label Tx: Enabled, Local Flow Label Rx: Enabled
Remote Flow Label Tx: Enabled, Remote Flow Label Rx: Enabled
Local Control Word: Enabled, Remote Control Word: Enabled
```

The following example shows how to configure a bridge domain on which flow label is enabled for the PW-related peers.

```
device# configure terminal
device(config)# bridge-domain 5
device(config-bridge-domain-5)# vc-id 8
device(config-bridge-domain-5)# logical-interface ethernet 0/5.15
device(config-bridge-domain-5)# logical-interface port-channel 2.200
device(config-bridge-domain-5)# peer 10.15.15.15 load-balance flow-label
device(config-bridge-domain-5)# peer 10.12.12.12 flow-label
device(config-bridge-domain-5)# peer 10.12.12.12 lsp lsp1 lsp2
device(config-bridge-domain-5)# local-switching
device(config-bridge-domain-5)# bpdu-drop-enable
device(config-bridge-domain-5)#
device(config-pw-pw_example)# exit
```

Configuring a static MAC address over an endpoint in a VPLS instance

You can configure a MAC address for a logical interface for an endpoint in a VPLS instance by completing the following task.

Note
Pre-configuration for the static mac is supported. Pre-configured static mac is shown as inactive mac.

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

```
device# configure terminal
```

2. Create the logical-interface (LIF) entry associated to a physical or port-channel interface, associate the LIF entry to the VPLS instance and configure the static mac associated with the LIF entry.

```
device(config)# mac-address-table static 0011.2345.6789 forward logical-interface
ethernet 0/2.200
```

3. Enter privileged EXEC mode.

```
device(config)# exit
```

4. (Optional) Verify the configuration using any of the following commands.

```
device# show mac-address-table
device# show mac-address-table static
device# show mac-address-table bridge-domain [bd-id]
```

Displaying MAC address information for VPLS bridge domains

- Enter the **show mac-address-table bridge-domain** command to display information about MAC addresses in VPLS bridge domains. The following example shows details of all MAC addresses learned on all bridge domains.

```
device# show mac-address-table bridge-domain all

VlanId/BD-Id   Mac-address           Type     State     Ports/LIF/peer-ip
629(B)            0011.2222.5555     Dynamic  Active    eth 1/3.100
629(B)            0011.2222.6666     Dynamic  Inactive  eth 1/1.500
629(B)            0011.2222.1122     Dynamic  Active    10.12.12.12
629(B)            0011.2222.3333     static   Inactive  po 5.700
629(B)            0011.0101.5555     Dynamic  Active    eth 1/2.400

Total MAC addresses : 5
```

- Enter the **show mac-address-table bridge-domain** command specifying a bridge domain to display information about MAC addresses for a specific bridge domain.

```
device# show mac-address-table bridge-domain 1

BD-Name   Mac-address     Type     State     Ports/LIF/Peer-IP
1         0011.2222.5555  Dynamic  Active    eth 1/3.200
1         0011.2222.6666  Dynamic  Inactive  eth 2/2.500

Total MAC addresses : 2
```

Configuring a VPLS instance

Prior to completing the following task, the underlying L3 configuration of MPLS tunnels must be completed. In addition, the logical interfaces to be attached the VPLS instance (bridge domain) should be configured. For information on configuring logical interfaces, refer to *Logical Interfaces*.
You can configure a VPLS instance by completing the following task.

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

```
device# configure terminal
```

2. Create a multipoint bridge domain.

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

   By default, the bridge-domain service type is multipoint. In this example, bridge domain 5 is configured as a multipoint service.
3. Configure a virtual connection identifier for the bridge domain.

```
device(config-bridge-domain-5)# vc-id 8
```

4. **Note**
   Logical interfaces representing bridge-domain endpoints must be created before they can be bound to a bridge domain.

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

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

   In this example, Ethernet logical interface 0/6.400 is bound to bridge domain 5.
5. Repeat Step 4 to bind other logical interfaces for attachment circuit endpoints to the bridge domain.

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

   In this example, port channel logical interface 2.200 is bound to bridge domain 5.
6. Configure peer IP addresses to create pseudowire (PW) endpoints.

   ```
   device(config-bridge-domain-5)# peer 10.15.15.15 load-balance
   ```

   In this example, a peer IP address of 10.15.15.15 is configured under bridge domain 5 and specifies load balancing.
7. Repeat Step 6 to configure more peer IP addresses to create PW endpoints.

   ```
   device(config-bridge-domain-5)# peer 10.12.12.12 lsp lsp1 lsp2
   ```

   In this example, a peer IP address of 10.12.12.12 under bridge domain 5 and specifies two label-switched paths (lsp1 and lsp2).
8. (Optional) Configure local switching for bridge domain 5.

   ```
   device(config-bridge-domain-5)# local-switching
   ```
9. (Optional) Enable dropping L2 bridge protocol data units (BPDUs) for bridge domain 5.

   ```
   device(config-bridge-domain-5)# bpdu-drop-enable
   ```
10. (Optional) Configure a PW profile under the bridge domain 5.

    ```
    device(config-bridge-domain-5)# pw-profile 2
    ```

The following example creates bridge domain 5 and configures virtual connection identifier 8 for the bridge domain. It binds ethernet and port-channel logical interfaces to the bridge domain and configures peer IP addresses under the domain. It configures local switching, enables dropping of L2 BPDUs, and configures a PW profile for the domain.

```
device# configure terminal
device(config)# bridge-domain 5
device(config-bridge-domain-5)# vc-id 8
device(config-bridge-domain-5)# logical-interface ethernet 0/6.400
device(config-bridge-domain-5)# logical-interface port-channel 2.200
device(config-bridge-domain-5)# peer 10.15.15.15 load-balance
device(config-bridge-domain-5)# peer 10.12.12.12 lsp lsp1 lsp2
device(config-bridge-domain-5)# local-switching
```

```
device(config-bridge-domain-5)# bpdu-drop-enable
device(config-bridge-domain-5)# pw-profile 2
```

Configuring a VLL instance

Prior to completing the following task, the Ethernet logical interface and pseudowire profiles must be created. There is an example at the end of this task that shows all the steps in order.
You can configure a VLL instance by completing the following task.

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

```
device# configure terminal
```

2. Create a point-to-point bridge domain to use VLL services.

```
device(config)# bridge-domain 3 p2p
```

   In this example, bridge domain 3 is created as a point-to-point service. By default, the bridge-domain service type is multipoint.
3. Configure a virtual connection identifier for the bridge domain.

```
device(config-bridge-domain-3)# vc-id 500
```

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

```
device(config-bridge-domain-3)# logical-interface ethernet 0/5.15
```

   In this example, the Ethernet logical interface 0/5.15 is bound to bridge domain 3.
5. Configure peer IP addresses to create pseudowire (PW) endpoints.

```
device(config-bridge-domain-3)# peer 10.10.10.10
```

6. Configure a PW profile under the bridge domain.

```
device(config-bridge-domain-3)# pw-profile to-mpls-nw
```

   The following example configures a PW profile 2 under bridge domain 3.
7. Repeat this configuration on the other peer device with appropriate parameters.
8. Enter Privileged EXEC mode.

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

9. (Optional) Display information about the configured VLL instance.

```
device# show bridge-domain 3

Bridge-domain Type: P2P , VC-ID: 3
Number of configured end-points:2 , Number of Active end-points: 2
VE if-indx: 0, Local switching: FALSE, bpdu-drop-enable: FALSE
PW-profile: default, mac-limit: 0
VLAN: 3, Tagged ports: 1(1 up), Un-tagged ports: 0 (0 up)
Tagged Ports: eth0/5.15
Un-tagged Ports:
Total VLL peers: 1 (1 Operational):
VC id: 3, Peer address: 10.10.10.10, State: Operational, uptime: 18 sec
Load-balance: True , Cos Enabled: False,
Tunnel cnt: 4
```

```
rsvp p105 (cos_enable:Falsecos_value:0)
rsvp p106 (cos_enable:Falsecos_value:0)
rsvp p107 (cos_enable:Falsecos_value:0)
rsvp p108 (cos_enable:Falsecos_value:0)
Assigned LSPs count:4 Assigned LSPs:p105 p106 p107 p108
Local VC lbl: 851968, Remote VC lbl: 985331,
Local VC MTU: 1600, Remote VC MTU: 1500,
Local VC-Type: 5, Remote VC-Type: 5
```

The following example shows the creation of a logical interface and a pseudowire profile in addition to the bridge domain and VLL instance configuration.

```
device# configure terminal
device(config)# interface ethernet 0/5
device(conf-if-eth-0/5)# switchport
device(conf-if-eth-0/5)# switchport mode trunk
device(conf-if-eth-0/5)# switchport trunk tag native-vlan
device(conf-if-eth-0/5)# shutdown
device(conf-if-eth-0/5)# logical-interface ethernet 1/5.15
device(conf-if-eth-lif-0/5.15)# vlan 200
device(conf-if-eth-lif-0/5.15)# exit
device(conf-if-eth-0/5)# exit
device(config)# pw-profile to-mpls-nw
device(config-pw-profile-to-mpls-nw)# mtu 1600
device(config-pw-profile-to-mpls-nw)# mtu-enforce true vc-mode tag
device(config-pw-profile-to-mpls-nw)# exit
device(config)# bridge-domain 3 p2p
device(config-bridge-domain-3)# vc-id 500
device(config-bridge-domain-3)# logical-interface ethernet 0/5.15
device(config-bridge-domain-3)# peer 10.10.10.10
device(config-bridge-domain-5)# pw-profile to-mpls-nw
```

Routing VE over VPLS

Virtual Private LAN Services (VPLS) enables you to connect remote sites over Multiprotocol Label Switching (MPLS) domain as if these sites were connected by a Layer 2 switch. It enables Virtual Circuits (VCs) to provide point-to-multipoint connection across the MPLS domain allowing traffic to flow between these remote sites on your Virtual Private Network (VPN).

Provider Edge equipments try to learn the MAC address of locally connected devices by flooding the broadcast and unknown unicast frames to other Provider Edge devices within the VPN. Associations are made between the remote MAC addresses and the

VC Lable Switched Paths (LSPs) used to reach these remote PE devices. Traffic is then routed over these learned LSPs for any frame for the destination PE devices.

Note

For SLX 9640, SLX 9740, and Extreme 8820 devices, enabling routing over BD for VE over VPLS is not supported when the pseudo wire profile on the bridge domain is in *Tag* mode. Due to this, the following will not be allowed on the CLI:

- Enabling routing on bridge domain which has pseudo wire profile mode set as *Tag*.
- Changing pseudo wire profile to *Tag* mode in case the pseudo wire profile is associated with any bridge domain that has routing enabled on it.

VE over VPLS will route packets between VPLS VE interface and all other IP interfaces outside of VPLS domain which reside on the PE devices. These include:

- Physical Interfaces.
- Other VLAN based VE interfaces for both tagged and un-tagged ports.
- VE interfaces which reside on other VPLS instances.

The following is an example of a complete configuration for VE over VPLS:

```
 ip proxy-arp
 ip address 15.15.15.15/24
 no shutdown
!
interface Ethernet 1/1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 100
 switchport trunk tag native-vlan
 no shutdown
 logical-interface ethernet 1/1.15
  vlan 15
 !
 bridge-domain 15 p2mp
 vc-id 15
 router-interface ve 15
 logical-interface ethernet 1/1.15
 pw-profile vplsPWprofile
 bpdu-drop-enable
 local-switching
!
pw-profile vplsPWprofile
 vc-mode tag
!
```

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

```
device# configure terminal
```

2. Create a broadcast bridge domain using the `bridge-domain <id> <type>` command with the <type> value set to *p2mp*.

```
device (config)# bridge-domain 15 p2mp
```

3. Configure a Virtual Connection Identifier (VC ID) for the bridge domain.

```
device (config-bridge-domain-15)# vc-id 15
```

4. Create a logical interface ID for them to be configured as Attachment Circuit (AC) endpoints.

```
device (config-bridge-domain-15)# interface ethernet 1/1/6.400
device (config-bridge-domain-15)# interface port-channel 2.200
```

5. Create the pseudo wire interface under the bridge domain.

```
device (config-bridge-domain-15)# peer 15.15.15.15 load-balance
device (config-bridge-domain-15)# peer 12.12.12.12 lsp lsp1 lsp2
```

6. From the global configuration mode, create a new pseudo wire profile. The pseudo wire profile is then applied to the VPLS instance.

```
device (config)# pw-profile vplsPWprofile
device (config-pw-profile-vplsPWprofile)#
```

7. Within the pseudo wire profile, set its Virtual connection (VC) mode to *Tagged*.

```
device (config-pw-profile-vplsPWprofile)# vc-mode tag
```

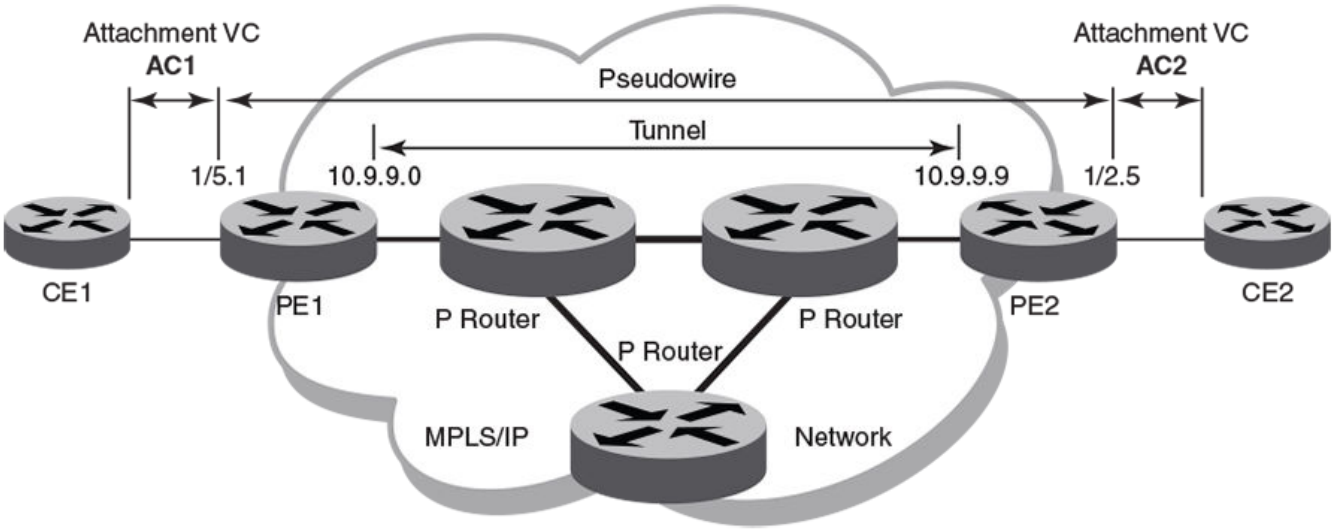
8. From withing the bridge domain context, apply the newly created pseudo wire profile to it.

```
device (config-bridge-domain-15)# pw-profile vplsPWprofile
```

9. Enable routing on the bridge domain by binding a router interface to the bridge domain.

```
device (config-bridge-domain-15)# router-interface ve 15
```

Configuration example for VPLS with switching between ACs and network core



The topology in the preceding figure shows a L2 VPN that enables transport of L2 traffic between two or more native Ethernet networks through an underlying Multiprotocol Label Switching (MPLS) provider network. Customer edge (CE) is the last mile and provider edge (PE) is the first mile node for packets transported towards the provider network. The provider intermediary network is an emulated switch (LAN) or wire (LINE) to the CE. The AC represents the logical link between the CE and PE.

Pseudowire is a circuit emulation infrastructure that extends L2 connectivity from CE1 to CE2 by way of PE1 and PE2. The tunnel is typically a L3 tunnel on which a L2 circuit is emulated.

The following examples show how to configure the provider edge devices (PE1 and PE2) shown in this topology.

Figure 24: VPLS configuration with switching between attachment circuits (ACs) and network core

PE1

```
device# configure terminal
device(config)# bridge-domain 500 p2mp
device(config-bridge-domain-500)# vc-id 501
device(config-bridge-domain-500)# peer 10.9.9.9 load-balance
device(config-bridge-domain-500)# logical-interface ethernet 0/5.1     ! AC1
device(config-bridge-domain-500)# exit

device(config)# pw-profile default
```

PE2

```
device# configure terminal
device(config)# bridge-domain 300 p2mp
device(config-bridge-domain-300)# vc-id 501
device(config-bridge-domain-300)# peer 10.9.9.0 load-balance
device(config-bridge-domain-500)# logical-interface ethernet 0/2.5     ! AC2
device(config-bridge-domain-500)# exit

device(config)# pw-profile default
```

VPLS MAC withdrawal

A MAC address withdrawal message is send with a MAC list Type Length Value (TLV). 200 MAC addresses are bulked and sent in one Mac TLV message.

Note
The MAC withdrawal support is only for explicit MAC addresses in MAC withdrawal TLV. Empty MAC list as well as sending MAC withdrawal TLV to specific subset of peers will not be supported.

The maximum number of MAC addresses supported is 5000 in a 5 second interval. The remaining MAC in the AC LIF are not sent. After the 5 second interval, another LIF down event triggers MAC withdrawal message for a new 5 second interval. MAC withdrawal is supported for both VPLS and MCT-VPLS. MPLS signals the MAC withdraw TLV to all the peers.

The **mac-address withdrawal** command enables MAC withdrawal on the bridge domain. The **no** form of the command disables MAC withdrawal.

Disabling MAC withdrawal on a bridge domain stops sending of MAC withdraw messages. MAC withdraw messages are received at the receiver end and MAC flush happens even when MAC address withdrawal is not enabled.

Enabling VPLS MAC address withdrawal

Convergence is faster when VPLS MAC address withdrawal is enabled.

Perform the following task to enable VPLS MAC address withdrawal.

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

   ```
   device# configure terminal
   ```

2. Enter bridge domain configuration mode. The following example shows how to enter configuration mode for bridge domain 1.

   ```
   device(config)# bridge domain 1
   ```

3. Enable VPLS MAC address withdrawal.

   ```
   device(config-bridge-domain-1)# mac-address withdrawal
   ```

   After it is enabled, use the **no** form of the command to disable MAC address withdrawal.

4. Return to privileged EXEC mode.

```
device(config-bridge-domain-1)# end
```

5. Verify the configuration.

```
device# show bridge-domain
Bridge-domain 1
-------------------------------
Bridge-domain Type: MP , VC-ID: 0
Number of configured end-points:  0 , Number of Active end-points: 0
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
MAC Withdrawal: Enabled
PW-profile: default, mac-limit: 0
Total VPLS peers: 0 (0 Operational):
device#
```

The following example shows how to enable VPLS MAC address withdrawal and then verify the configuration for bridge domain 1.

```
device# configure terminal
device(config)# bridge domain 1
device(config-bridge-domain-1)# mac-address withdrawal
device(config-bridge-domain-1)# end
!
device# show bridge-domain
Bridge-domain 1
-------------------------------
Bridge-domain Type: MP , VC-ID: 0
Number of configured end-points:  0 , Number of Active end-points: 0
VE if-indx: 0, Local switching: TRUE, bpdu-drop-enable: TRUE
MAC Withdrawal: Enabled
PW-profile: default, mac-limit: 0
Total VPLS peers: 0 (0 Operational):
device#
```

MAC Movement Detection and Resolution

MAC Movement Overview on page 174

MAC Movement Detection and Resolution is a new and improved mechanism to prevent loop detection in networks. This feature has advantages over the existing Extreme proprietary Loop Detection feature.

Note

- Repeated MAC Movement Detection and Resolution will work for switch-port.
- Loop detection using MAC Movement Detection is mutually exclusive of STP and ELD protocol operations.
- MAC Movement Detection is not considered for a port where port-security is enabled and a restrict violation action is triggered.

MAC Movement Overview

A MAC address is considered to have been moved when the same MAC address is received on a different interface in the same VLAN. In MCT, MAC movement is allowed on both local and remote nodes. However, a high MAC move rate can indicate the existence of a loop in the network, or an issue with the server-side interface, resulting in flapping. A high move rate requires the control plane to process an extremely high rate of MAC learn events and can potentially exhaust the control plane resources.

MAC Move Definitions

- Rapid MAC Movement: A MAC that moves across multiple Logical Interfaces (LIF), ports, and VLANs is tracked for each second it moves. If the number of moves crosses the defined threshold, then it is treated as a MAC Move violation.
- Slow MAC Movement: In the first second that MAC movement is detected, the movement is monitored. In that first second, if the number of moves does not cross the threshold, then the MAC will be tracked for a maximum of 10 seconds. If after 10 seconds the total number of moves is within the threshold, then no action is taken. If it exceeds the threshold limit, then an action will be triggered.

MAC Movement Detection

Once mac-movement detection is enabled and configured, MAC movement is monitored. If number of MAC moves in one second is more than the (user-defined) threshold limit, three values (old LIF, current LIF, number of MAC moves) are recorded.

This list of recorded values is parsed automatically; if a port is determined to have a high number LIFs that are experiencing MAC moves, then those LIFs are acted upon (RASlog is the default action, or shutdown).

For example, MAC A is moving between port 1, vlan 10 and port 2, vlan 10. MAC B is moving between port 1, vlan 20 and port 3, vlan 20. MAC C is moving between port 4, vlan 10 and port 5, vlan 10. All MACs have crossed the (user-defined) threshold. The list looks like this:

- MAC A: port 1,vlan 10; and port 2, vlan 10
- MAC B: port 1, vlan 20; and port 3, vlan 20
- MAC C: port 4, vlan 10; and port 5, vlan 10

MAC Movement Resolution

Based on the example above, Port 1 is selected for the mac-movement action (`shutdown | raslog`) because it has a higher number of LIFs in the list. The LIFs port 1,vlan 10 and port 1, vlan 20 are either shutdown with the actions logged, or the movement details are committed to the RAS log.

The default mac-movement action is `raslog`. All mac-movement information is saved to the RAS log. When the auto-recovery time limit expires, the LIF becomes operational, and the tracking process is started again.

Note

Logical interfaces (LIFs) that are part of a tunnel, inter-chassis link (ICL), or pseudowire (PW) are ignored and are not subject to any resolution action.

Auto-recovery is enabled by default. The LIF will remain shut down for 5 minutes (default). This time is configurable to a range of 3 through 30 minutes. The auto-recovery mode can be disabled from the command line.

Shutdown can be disabled from the command line, and logging enabled as the only action when the threshold is exceeded. The MAC movement details will be captured in the RAS log.

RAS Log Examples:

- MAC move detected and action defined as RAS log
  - 2019/12/09-02:38:37, [L2SS-1034], 16815, DCE, ERROR, SLX, Repeated MAC move detected for MAC 0000.0500.0001 VLAN 10 on Interface Eth 0/18
  - 2019/12/09-02:38:37, [L2SS-1035], 16816, DCE, ERROR, SLX, Repeated mac move detected for MAC 0000.0500.0001 BRIDGE-DOMAIN 11 on Interface Eth 0/18.11

- MAC move detected and action to shutdown LIF
  - 2019/12/09-02:32:02, [L2SS-1033], 16805, DCE, INFO, SLX, Shut down recovery(from MAC Move detect) for Logical- Interface Eth 0/18.11 and BRIDGE-DOMAIN 11
  - 2019/12/09-02:32:02, [L2SS-1025], 16806, DCE, INFO, SLX, Shut down recovery(from MAC Move detect) for Interface Eth 0/18 and VLAN 10
- Shutdown recovery
  - 2019/12/09-02:32:12, [L2SS-1024], 16807, DCE, ERROR, SLX, Repeated MAC move detected for MAC 0000.0500.0001 VLAN 10, Interface Ethernet 0/6 shut down for VLAN 10
  - 2019/12/09-02:32:12, [L2SS-1032], 16808, DCE, ERROR, SLX, Repeated MAC move detected for MAC 0000.0500.0001 BRIDGE-DOMAIN 11, interface Ethernet 0/6 shut down

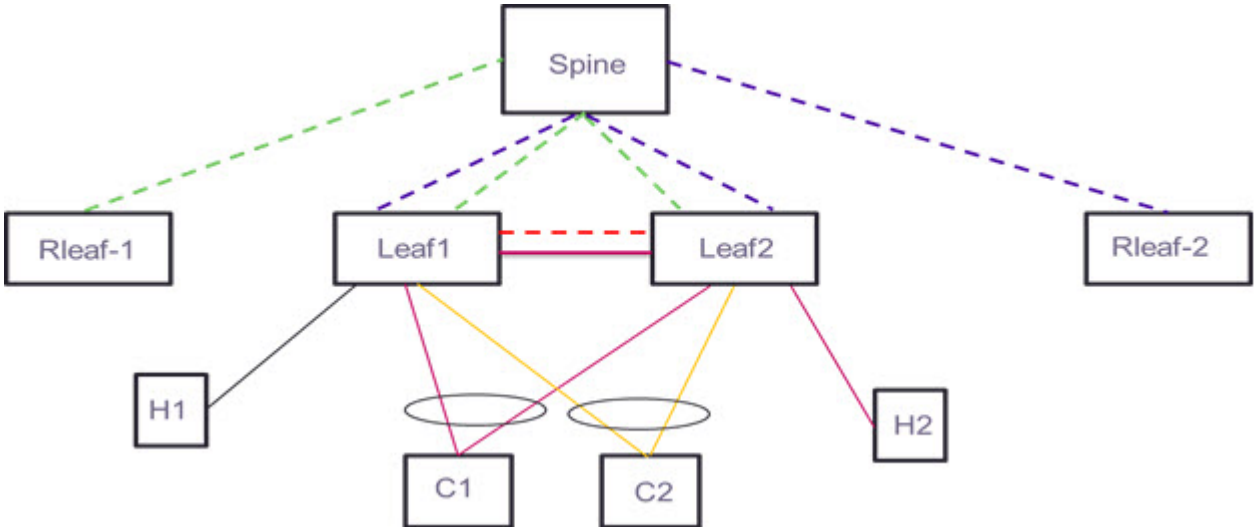


Figure 25: MAC Movement Detection Scenario

MCT Use Case: MAC loop between CCEPs

If MAC movement between C1 and C2 has crossed the threshold MAC move limit, the LIF shutdown action is taken on the cluster node with the higher IP address. For example: Leaf1 has the higher cluster IP configured. Leaf1 takes the decision to shut down the LIF connecting to C1 and communicates to Leaf2 to shutdown the corresponding LIF connecting to C1.

MCT Use Case: MAC loop between CCEP and CEP

If MAC movement between C1 and H1 has crossed the threshold MAC move limit the LIF shutdown action is taken on the cluster node with the higher IP address. For example, Leaf1 has the higher cluster IP configured. Leaf1 takes the decision to shut down the LIF on C1, and communicates to Leaf2 to shutdown the corresponding LIF on C1. If the MAC movement is happening between the CCEP and CEP, the action will always be taken on the CCEP.

MCT Use Case: MAC loop between CEPs

If MAC movement between H1 and H2 has crossed the threshold MAC move limit, the action is taken by each Leaf independently. Leaf1 and Leaf2 will run MAC move detection separately.

Note
In an IP fabric configuration, it may be necessary for the MAC movement detection feature to take precedence over BGP dampening. In this situation, the BGP mac dampening count must be set at lower value than the mac-move threshold, as shown below.

```
device(config)# do show running-config evpn
    evpn a duplicate-mac-timer 5 max-count 10
    !
    device(config)# exit
    device(config)# show mac-address-table mac-move
    Mac Move Action: Shutdown
    Mac Move detect: Enable
    Threshold: 7
    Auto-Recovery: Enable
    Auto-Recovery-time: 3
```

Note
In an IP fabric configuration, this feature must be enabled on all leaf nodes.

Note
This feature is also supported on bridge domains.

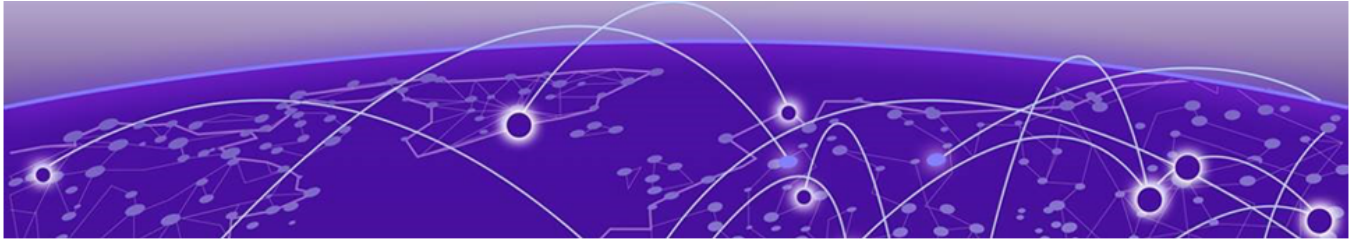
MAC movement Detection and Resolution Commands

- mac-address-table mac-move detect
- mac-address-table mac-move limit
- mac-address-table mac-move action { shutdown | raslog }
- mac-address-table mac-move auto-recovery enable { time } [minutes]
- clear mac-address-table mac-move shut-list
- show bridge-domain [bd-id] logical-interface
- show mac-address-table mac-move shut list
- show mac-address-table mac-move
- show vlan

- show vlan detail
- show vlan interface port-channel

Table 36: MAC movement detection and resolution commands

Command	Details
mac-address-table mac-move detect	Enables MAC movement detection. Disabled by default. Once enabled, use the no form to disable.
mac-address-table mac-move limit	Defines the movement threshold. Enter a value; default is 20, range is 3 through 500.
mac-address-table mac-move action	Define the action to be taken when MAC movement exceeds the specified threshold. Default action is RASlog. Enabled by default when mac-address-table mac-move detect is enabled.
mac-address-table mac-move auto-recovery enable	Enables auto-recovery on the port/interface after a shutdown action. Enabled by default when shutdown is enabled.
mac-address-table mac-move auto-recovery time	Sets the auto recovery time. Default auto-reovery time is 5 minutes. Range is 3 through 30 minutes.
clear mac-address-table mac-move shut-list	Clears all entries from the shutdown list.
show bridge-domain [bd-id] logical-interface	LIF output has been modified to indicate when the LIF is down due to mac-movement.
show mac-address-table mac-move shut list	Displays information for all entries in the shutdown list.
show mac-address-table mac-move	Displays all mac-move configurations.
show vlan	Output has been modified to indicate when the LIF/port is down due to mac-movement.
show vlan detail	Output has been modified to indicate when the LIF/port is down due to mac-movement.
show vlan interface port-channel	Output has been modified to indicate when the VLAN is down due to mac-movement.



802.1ag Connectivity Fault Management

Enabling or disabling CFM on page 183
Creating a Maintenance Domain on page 183
Creating and configuring a Maintenance Association on page 183
Displaying CFM configurations on page 184

Bridges are increasingly used in networks operated by multiple independent organizations, each with restricted management access to each other's equipment. CFM provides capabilities for detecting, verifying and isolating connectivity failures in such networks.

There are multiple organizations involved in a Metro Ethernet Service: Customers, Service Providers and Operators.

Customers purchase Ethernet Service from Service Providers. Service Providers may utilize their own networks, or the networks of other Operators to provide connectivity for the requested service. Customers themselves may be Service Providers, for example a Customer may be an Internet Service Provider which sells Internet connectivity.

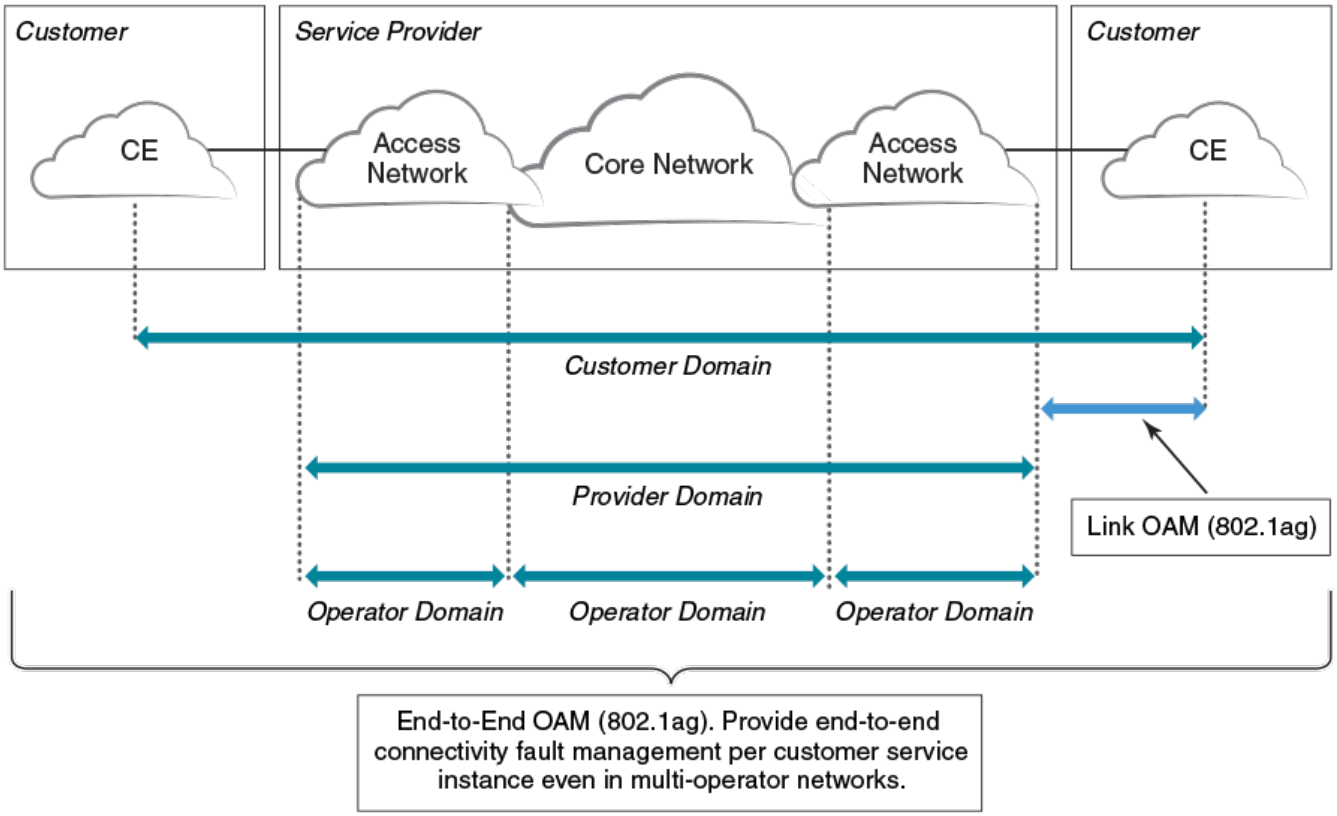


Figure 26: OAM Ethernet tools

Maintenance Domain (MD)

A Maintenance Domain is part of a network controlled by a single operator. In the following figure, a customer domain, provider domain and operator domain are described.

The Maintenance Domain (MD) levels are carried on all CFM frames to identify different domains. For example, in the following figure, some bridges belong to multiple domains. Each domain associates to an MD level.

- Customer Level: 5-7
- Provider Level: 3-4
- Operator Level: 0-2

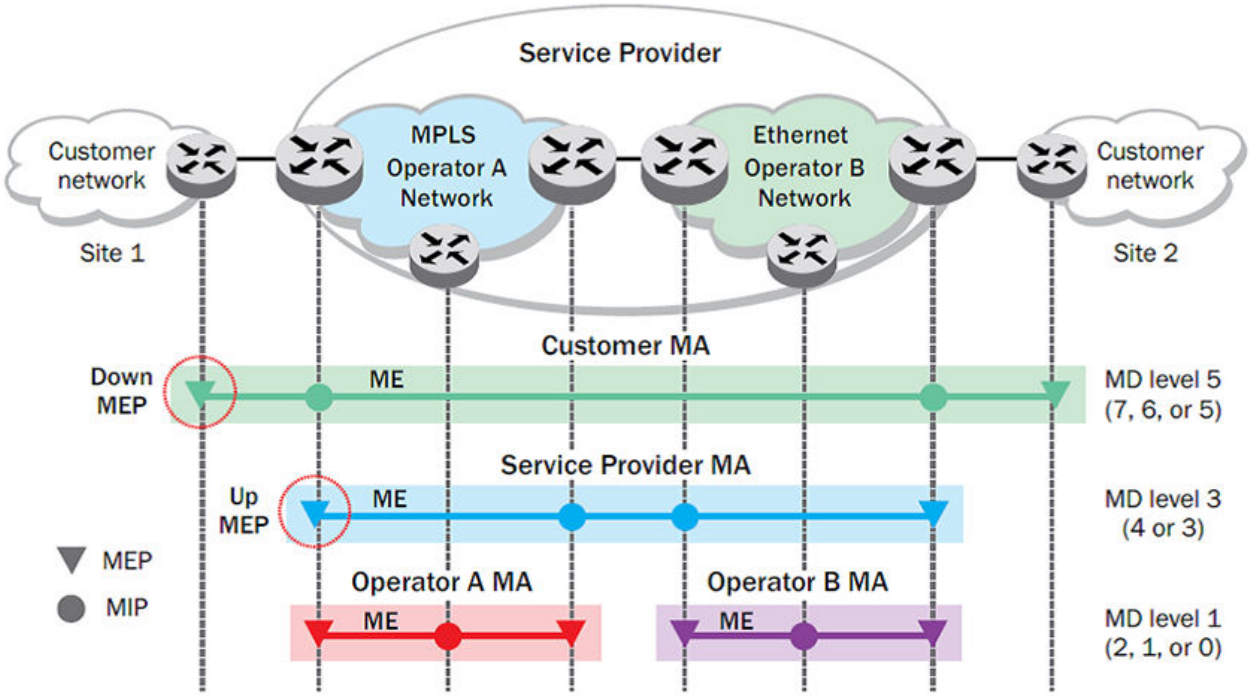


Figure 27: CFM deployment

Maintenance Association (MA)

Every MD can be further divided into smaller networks having multiple Maintenance End Points (MEP). Usually an MA is associated with a service instance (for example, a VLAN or a VPLS).

Maintenance End Point (MEP)

An MEP is located on the edge of an MA and defines the endpoint of the MA. Each MEP has unique ID (MEPID) within the MA. The connectivity in a MA is defined as connectivity between MEPs. MEPs generate a Continuity Check Messages that are multicast to all other MEPs in same MA to verify the connectivity.

Each MEP has a direction, down or up. Down MEPs receive CFM PDUs from the LAN and sends CFM PDUs towards the LAN. Up MEPs receive CFM PDUs from a bridge relay entity and sends CFM PDUs towards the bridge relay entity on a bridge. End stations support down MEPs only, as they have no bridge relay entities.

Maintenance Intermediate Point (MIP)

An MIP is located within a MA. It responds to Loopback and Linktrace messages for Fault isolation.

CFM Hierarchy

MD levels create a hierarchy in which 802.1ag messages sent by customer, service provider, and operators are processed by MIPs and MEPs at the respective level of the message. A common practice is for the service provider to set up a MIP at the customer MD level at the edge of the network, as shown in the figure above, to allow the customer to check continuity of the Ethernet service to the edge of the network. Similarly, operators set up MIPs at the service provider level at the edge of their respective networks, as shown in the figure above, to allow service providers to check the continuity of the Ethernet service to the edge of the operators' networks. Inside an operator network, all MIPs are at the respective operator level, also shown in the figure above.

Mechanisms of Ethernet IEEE 802.1ag OAM

Mechanisms supported by IEEE 802.1ag include Connectivity Check (CC), Loopback, and Link trace. Connectivity Fault Management allows for end-to-end fault management that is generally reactive (through Loopback and Link trace messages) and connectivity verification that is proactive (through Connectivity Check messages).

Fault detection (continuity check message)

Each MEP transmits periodic multicast CCMs towards other MEPs. For each MEP, there is 1 transmission and n-1 receptions per time period. Each MEP has a remote MEP database. It records the MAC address of remote MEPs.

Fault verification (Loopback messages)

A unicast Loopback Message is used for fault verification. A Loopback message helps a MEP identify the precise fault location along a given MA. A Loopback message is issued by a MEP to a given MIP along an MA. The appropriate MIP in front of the fault responds with a Loopback reply. The MIP behind the fault do not respond. For Loopback to work, the MEP must know the MAC address of the MIP to ping.

Fault isolation (Linktrace messages)

Linktrace mechanism is used to isolate faults at Ethernet MAC layer. Linktrace can be used to isolate a fault associated with a given Virtual Bridge LAN Service. Note that fault isolation in a connectionless (multi-point) environment is more challenging than a connection oriented (point-to-point) environment. In case of Ethernet, fault isolation can be even more challenging since a MAC address can age out when a fault isolates the MAC address. Consequently a network-isolating fault results in erasure of information needed for locating the fault.

Enabling or disabling CFM

To enable or disable the Connectivity Fault Management (CFM) protocol globally on the devices and enter into the CFM protocol configuration mode, enter the following command.

```
device# configure terminal
device(config)# protocol cfm
device(config-cfm)#
```

The **no** form of the command disables the CFM protocol.

Creating a Maintenance Domain

A Maintenance Domain (MD) is the network or the part of the network for which faults in connectivity are to be managed. A Maintenance Domain consists of a set of Domain Service Access Points.

An MD is fully connected internally. A Domain Service Access Point associated with an MD has connectivity to every other Domain Service Access Point in the MD, in the absence of faults.

Each MD can be separately administered.

The **domain-name** command in Connectivity Fault Management (CFM) protocol configuration mode creates a maintenance domain with a specified level, name, and ID and enters the specific MD mode specified in the command argument.

```
device# configure terminal
device(config)# protocol cfm
device(config-cfm)# domain-name md1 id 1 level 4
device(config-cfm-md-md1)#
```

The **no** form of the command removes the specified domain from the CFM protocol configuration mode.

Creating and configuring a Maintenance Association

1. Create a MA within a specific domain, use the **ma-name** command.

   ```
   device# configure terminal
   device(config)# protocol cfm
   device(config-cfm)# domain name md1 id 1 level 4
   device(config-cfm-md-md1)# ma-name ma1 id 1 vlan-id 30 priority 4
   device(config-cfm-md-ma-ma1)#
   ```

   This command changes the Maintenance Domain (MD) mode to the specific MA mode.
2. Set the time interval between two successive Continuity Check Messages (CCMs) that are sent by Maintenance End Points (MEP) in the specified MA, use the **ccm-interval** command.

   ```
   device# configure terminal
   device(config)# protocol cfm
   device(config-cfm)# domain name md1 id 1  level 4
   device(config-cfm-md-md1)# ma-name ma1 id 1 vlan-id 30 priority 3
   ```

```
device(config-cfm-md-ma-ma1)# ccm-interval 10-second
device(config-cfm-md-ma-ma1)#
```

The **id** field specifies the short MAID format that is carried in the CCM frame. The default time interval is 10 seconds.

3. Add local ports as MEP to a specific maintenance association using the **mep** command in MA mode.

```
device# configure terminal
device(config)# protocol cfm
device(config-cfm)# domain name md1 id 1 level 4
device(config-cfm-md-md1)# ma-name ma1 id 1 vlan-id 30 priority 3
device(config-cfm-md-ma-ma1)# mep 1 down ethernet 1/2
device(config-cfm-md-ma-mep-1)#
```

To configure a CFM packet to a **Down MEP**, you must sent it out on the port on which it was configured. To configure a Connectivity Fault Management (CFM) packet to an **Up MEP**, you must sent it to the entire VLAN for multicast traffic and the unicast traffic must be sent to a particular port as per the MAC table.

4. Configure the remote MEPs using the **remote-mep** command.

```
device# configure terminal
device(config)# protocol cfm
device(config-cfm)# domain name md1 id 1clevel 4
device(config-cfm-md-md1)# ma-name ma1 id 1 vlan-id 30 priority 3
device(config-cfm-md-ma-ma1)# mep 1 down ethernet 1/2
device(config-cfm-md-ma-mep-1)# remote-mep 2
device(config-cfm-md-ma-mep-1)#
```

If a remote MEP is not specified, the remote MEP database is built based on the CCM. If one remote MEP never sends CCM, the failure cannot be detected.

5. Configure the conditions to automatically create MIPs on ports using the **mip-policy** command, in Maintenance Association mode.

```
device# configure terminal
device(config)# protocol cfm
device(config-cfm)# domain name md1 id 1 level 4
device(config-cfm-md-md1)#ma-name ma1 id 1 vlan-id 30 pri 7
device(config-cfm-md-ma-ma1)#mip-policy explicit
device(config-cfm-md-ma-ma1)#
```

A MIP can be created on a port and VLAN, only when explicit or default policy is defined for them. For a specific port and VLAN, a MIP is created at the lowest level. Additionally, the level created should be the immediate higher level than the MEP level defined for this port and VLAN.

Displaying CFM configurations

The following commands are used to display the CFM configurations and connectivity status.

show cfm

Use the **show cfm** command to display the Connectivity Fault Management (CFM) configuration.

```
device# show cfm
```

```
Domain: md1
Index: 1
Level: 7
  Maintenance association: ma5
  MA Index: 5
  CCM interval: 100 ms
  Bridge-Domain ID: 50
  Priority: 7
  MAID Format: Short
  MEP   Direction  MAC              PORT       VLAN         INNER-VLAN   PORT-STATUS-TLV
  ====  =========  =========        ====       ====         ==========   ===============
  1     UP         609c.9f5f.700d   Eth 1/9    50           --           N
```

Note
For the **show cfm** command to generate output, you must first enable CFM in protocol configuration mode.

show cfm connectivity

Use the **show cfm connectivity** command to display the Connectivity Fault Management (CFM) configuration.

The following commands display the received port status tlv state at RMEP.

```
device# show cfm connectivity
Domain: md1
Index: 1
Level: 7
  Maintenance association: ma5
  MA Index: 5
  CCM interval: 100 ms
  Bridge-Domain ID: 50
  Priority: 7
  MAID Format: Short
    MEP Id: 1
    MEP Port: Eth 1/9
      RMEP  MAC             VLAN/PEER         INNER-VLAN    PORT      STATE
      ====  ===             =========         ==========    ====      =====
      2     609c.9f5e.4809  19.1.1.1           --            --       OK
```

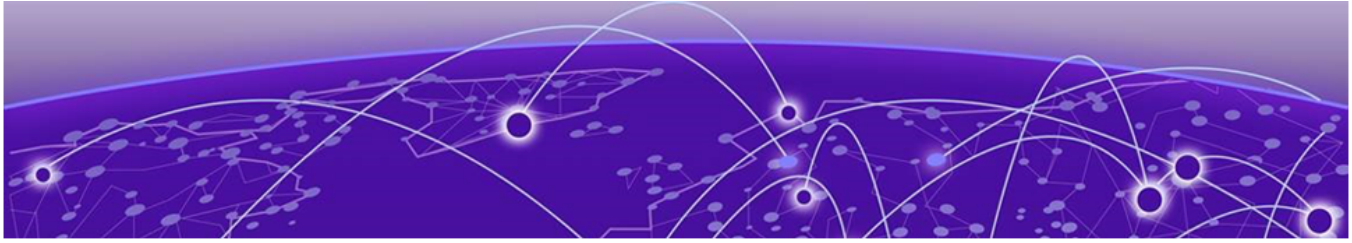
Note
For the **show cfm** command to generate output, you must first enable CFM in protocol configuration mode.

show cfm brief

Use the **show cfm brief** command to display the Connectivity Fault Management (CFM) brief output.

```
device# show cfm brief
Domain: md1
Index: 1
Level: 7  Num of MA: 1
  Maintenance association: ma5
  MA Index: 5
  CCM interval: 100 ms
  Bridge-Domain ID: 50
```

```
Priority: 7
MAID Format: Short
Num of MEP: 1  Num of RMEP: 1
rmepfail: 0 rmepok: 1
```



802.1d Spanning Tree Protocol

Spanning Tree Protocol overview on page 187
Spanning Tree Protocol configuration notes on page 187
STP features on page 191
STP parameters on page 192
Configuring STP on page 195

Spanning Tree Protocol overview

The IEEE 802.1d Spanning Tree Protocol (STP) runs on bridges and switches that are 802.1d-compliant.

These variants are Rapid STP (RSTP), Multiple STP (MSTP), Per-VLAN Spanning Tree Plus (PVST+), and Rapid-PVST+ (R-PVST+)

When the spanning tree algorithm is run, the network switches transform the real network topology into a spanning tree topology. In an STP topology any LAN in the network can be reached from any other LAN through a unique path. The network switches recalculate a new spanning tree topology whenever there is a change to the network topology.

For each LAN, the switches that attach to the LAN select a designated switch that is the closest to the root switch. The designated switch forwards all traffic to and from the LAN. The port on the designated switch that connects to the LAN is called the designated port. The switches decide which of their ports is part of the spanning tree. A port is included in the spanning tree if it is a root port or a designated port.

STP runs one spanning tree instance (unaware of VLANs) and relies on long duration forward-delay timers for port state transition between disabled, blocking, listening, learning and forwarding states.

Spanning Tree Protocol configuration notes

The Extreme device supports STP as described in the IEEE 802.1d-1998 specification.

The STP is disabled by default on the Extreme device. Thus, any new VLANs you configure on the Extreme device have STP disabled by default.

Optional features

The following STP configuration features are optional:

- Root guard
- BPDU guard
- PortFast

STP states

A network topology of bridges typically contains redundant connections to provide alternate paths in case of link failures. The redundant connections create a potential for loops in the system. As there is no concept of time to live (TTL) in Ethernet frames, a situation may arise where there is a permanent circulation of frames when the network contains loops. To prevent this, a spanning tree connecting all the bridges is formed in real time.

Every Layer 2 interface running the STP is in one of these states:

State	Action or inaction
Blocking	The interface does not forward frames. Redundant ports are put in a blocking state and enabled when required. This is a transitional state after initialization.
Listening	The interface is identified by the spanning tree as one that should participate in frame forwarding. This is a transitional state after the blocking state for a legacy STP.
Learning	The interface prepares to participate in frame forwarding. This is a transitional state after the blocking state for a legacy STP.
Forwarding	The interface forwards frames. This is a transitional state after the learning state.
Disabled	The interface is not participating in a spanning tree because of shutdown of a port or the port is not operationally up. Any of the other states may transition into this state.

BPDUs

To construct a spanning tree requires knowledge of the all the participants. The bridges must determine the root bridge and compute the port roles (root, designated, or blocked) with only the information that they have. To ensure that each bridge has enough information, the bridges use BPDUs to exchange information about bridge IDs and root path costs.

A bridge sends a BPDU frame using the unique MAC address of the port itself as a source address, and a destination address of the STP multicast address 01:80:C2:00:00:00.

BPDUs are exchanged regularly (every 2 seconds by default) and enable switches to keep track of network changes and to start and stop forwarding through ports as required.

When a device is first attached to a switch port, it does not immediately forward data. It instead goes through a number of states while it processes inbound BPDUs and determines the topology of the network. When a host is attached, after a listening and learning delay of about 30 seconds, the port always goes into the forwarding state. The time spent in the listening and learning states is determined by the forward delay. However, if instead another switch is connected, the port may remain in blocking mode if it would cause a loop in the network.

There are four types of BPDUs in the original STP specification:

- Configuration BPDU (CBPDU) is used for spanning tree computation.
- Topology Change Notification (TCN) BPDU is used to announce changes in the network topology.
- RSTP BPDU is used for RSTP
- MSTP BPDU is used for MSTP

TCN BPDUs

TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

Consider these configuration rules:

- TCN BPDUs are sent per VLAN.
- TCN BPDUs are sent only in those VLANs in which a topology change is detected.
- TCN BPDUs are sent only in those VLANs for which the bridge is not the root bridge.
- If a topology change is detected on a VLAN for which the bridge is the root bridge, the topology change flag is set in the configuration BPDU that is sent out.

For a given link, in conjunction with the configuration rules, a TCN BPDU is sent out as follows:

- On an access port, only a standard IEEE TCN BPDU is sent out. This TCN BPDU corresponds to a topology change in the access VLAN.
- On a trunk port, if VLAN 1 is allowed (either untagged or untagged), a standard IEEE TCN BPDU is sent for VLAN 1.
- On a trunk port, if the native VLAN is not 1, an untagged TCN BPDU is sent to Cisco or Extreme proprietary MAC address for that VLAN.
- On a trunk port, a tagged TCN BPDU is sent to Cisco or Extreme proprietary MAC address for a tagged VLAN.

As part of the response to TCN BPDUs, the Topology Change and Topology Change Acknowledgment flags are set in all configuration BPDUs corresponding to the VLAN for which the TCN was received.

When a topology change is detected on a trunk port, it is similar to detecting topology changes in each VLAN that is allowed on that trunk port. TCN BPDUs are sent for each VLAN as per the rules.

STP configuration guidelines and restrictions

- Only one form of a standing tree protocol, such as STP or RSTP, can be enabled at a time. You must disable one form of xSTP before enabling another.
- When any form of STP is enabled globally, that form of STP is enabled by default on all switch ports.
- LAGs are treated as normal links for any form of STP.
- The STP is disabled by default on the SLX device. Thus, any new VLANs you configure on the SLX device have STP disabled by default.
- PVST/RPVST BPDUs are flooded only if PVST/RPVST is not enabled. STP/RSTP (IEEE) BPDUs are never flooded if STP/RSTP is not enabled.

Understanding the default STP configuration

Table 37: 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 the switch defaults for the interface-specific configuration.

Table 38: Default interface specific configuration

Parameter	Default setting
Spanning tree	Enabled on the interface
Automatic edge detection	Disabled
Path cost	2000
Edge port	Disabled
Guard root	Disabled
Hello time	2 seconds
Link type	Point-to-point
Portfast	Disabled
Port priority	128
BPDU restriction	Restriction is disabled.

STP features

The following sections discuss root guard, BPDU guard, and PortFast.

Root guard

At times it is necessary to protect the root bridge from malicious attack or even unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge, causing severe bottlenecks in the data path. These types of mistakes or attacks can be avoided by configuring root guard on ports of the root bridge.

The root guard feature provides a way to enforce the root bridge placement in the network and allows STP and its variants to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires. Errors are triggered if any change from the root bridge placement is detected.

When root guard is enabled on a port, it keeps the port in designated FORWARDING state. If the port receives a superior BPDU, which is a root guard violation, it sets the port into a DISCARDING state and triggers a Syslog message and an SNMP trap. No further traffic will be forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or wrongly configured STP or RSTP bridges.

Root guard should be configured on all ports where the root bridge should not appear. In this way, the core bridged network can be cut off from the user network by establishing a protective perimeter around it.

Once the port stops receiving superior BPDUs, root guard automatically sets the port back to a FORWARDING state after the timeout period has expired.

BPDU guard

In a valid configuration, edge port-configured interfaces do not receive BPDUs. If an edge port-configured interface receives a BPDU, an invalid configuration exists, such as the connection of an unauthorized device. The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service.

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Extreme device port to which the end station is connected. The STP BPDU guard shuts down the port and puts it into an "error disabled" state. This disables the connected device's ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a message is displayed to warn the network administrator of an invalid configuration.

The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service with the **no shutdown** command if error disable recovery is not enabled by enabling the **errdisable-timeout** command. The interface can also be automatically configured to be enabled after a timeout. However, if the offending BPDUs are still being received, the port is disabled again.

Expected behavior in an interface context

When BPDU Guard is enabled on an interface, the device is expected to put the interface in Error Disabled state when BPDU is received on the port when edge-port and BPDU guard is enabled on the switch interface. When the port ceases to receive the BPDUs, it does not automatically switch to edge port mode, you must configure **error disable timeout** or **no shutdown** on the port to move the port back into edge port mode.

Error disable recovery

A port is placed into an error-disabled state when:

- A BPDU guard violation or loop detection violation occurs
- The number of inError packets exceeds the configured threshold
- An EFM-OAM enabled interface receives a critical event from the remote device (functionally equivalent to a disable state)

Once in an error disable state, the port remains in that state until it is re-enabled automatically or manually.

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, you can specify the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds. By default, the timeout feature is disabled.

PortFast

Consider the following when configuring PortFast:

- Do not enable PortFast on ports that connect to other devices.
- PortFast only needs to be enabled on ports that connect to workstations or PCs. Repeat this configuration for every port connected to workstations or PCs.
- Enabling PortFast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.
- If BPDUs are received on a PortFast- enabled interface, the interface loses the edge port status unless it receives a **shutdown**/**no shutdown** command.
- PortFast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

STP parameters

The following section discusses bridge parameters.

Bridge parameters

These parameters are set in STP, RSTP, MSTP, PVST+, and R-PVST+.

Bridge priority

Use this parameter to specify the priority of a device and to determine the root bridge.

Each device has a unique bridge identifier called the bridge ID. The bridge ID is an 8 byte value that is composed of two fields: a 2 B bridge priority field and the 6 B MAC address field. The value for the bridge priority ranges from 0 to 61440 in increments of 4096. The default value for the bridge priority is 32768. You use the **bridge-priority** command to set the appropriate values to designate a device as the root bridge or root device. A default bridge ID may appear as `32768.768e.f805.5800`. If the bridge priorities are equal, the device with the lowest MAC address is elected the root.

After you decide what device to designate as the root, you set the appropriate device bridge priorities. The device with the lowest bridge priority becomes the root device. When a device has a bridge priority that is lower than that of all the other devices, it is automatically selected as the root.

The root device should be centrally located and not in a "disruptive" location. Backbone devices typically serve as the root because they usually do not connect to end stations. All other decisions in the network, such as which port to block and which port to put in forwarding mode, are made from the perspective of the root device.

You may also specify the bridge priority for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge Protocol data units (BPDUs) carry information between devices. All the devices in the Layer 2 network, participating in any variety of STP, gather information on other devices in the network through an exchange of BPDUs. As the result of exchange of the BPDUs, the device with the lowest bridge ID is elected as the root bridge

When setting the bridge forward delay, bridge maximum aging time, and the hello time parameters keep in mind that the following relationship should be kept:

```
(2 × (forward-delay − 1)) ≥ max-age ≥ (2 ×(hello-time + 1))
```

Bridge forward delay

The bridge forward delay parameter specifies how long an interface remains in the listening and learning states before the interface begins forwarding all spanning tree instances. The valid range is from 4 through 30 seconds. The default is 15 seconds.

Additionally, you may specify the forward delay for a specific VLAN. If the VLAN parameter is not provided, the bridge forward delay value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge maximum aging time

You can use this setting to configure the maximum length of time that passes before an interface saves its BPDU configuration information.

Keeping with the inequality shown above, when configuring the maximum aging time, you must set the value greater than the hello time. The range of values is 6 through 40 seconds while he default is 20 seconds.

You may specify the maximum aging for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Bridge hello time

You can use this parameter to set how often the device interface broadcasts hello BPDUs to other devices.

Use the **`hello-time`** command to configure the bridge hello time. The range is from 1 through 10 seconds. The default is 2 seconds.

You may also specify the hello time for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

Error disable timeout parameter

These parameters are be set in STP, RSTP, MSTP, PVST+, and R-PVST+.

When the STP BPDU guard disables a port, the port remains in the disabled state unless the port is enabled manually. The parameter specifies the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds.

By default, the timeout feature is disabled.

Port-channel path cost parameter

This parameter can be set in STP, RSTP, MSTP, PVST+, and R-PVST+ mode.

There are two path cost options:

- Custom - Specifies that the path cost changes according to the port channel bandwidth.
- Standard - Specifies that the path cost does not change according to the port channel bandwidth.

The default port cost is standard.

Configuring STP

The following section discusses configuring STP.

Enabling and configuring STP globally

You can enable STP or STP with one or more parameters enabled.

The parameters can be configured individually by:

1. Entering the commands in steps 1 and 2
2. Running the relevant parameter command
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable STP globally.

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

A spanning tree can be disabled by entering the **no protocol spanning-tree stp** command.

3. Describe or name the STP.

```
device(config-stp)# description stp1
```

A description is not required.

4. Specify the bridge priority.

```
device(config-stp)# bridge-priority 4096
```

The bridge with the lowest priority number (highest priority) is designated the root bridge . The range of values is 0 through 61440; values can be set only in increments of 4096. The default priority is 32678.

5. Specify the bridge forward delay.

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

The forward delay specifies how long an interface remains in the listening and learning states before it begins forwarding all spanning tree instances. The valid range is from 4 through 30 seconds. The default is 15 seconds.

6. Configure the maximum aging time.

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

   This parameter controls the maximum length of time that passes before an interface saves its BPDU configuration information. You must set the maximum age to be greater than the hello time. The range is 6 through 40 seconds. The default is 20 seconds.

7. Configure the maximum hello time.

```
device(config-stp)# hello-time 8
```

   The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The default is 2 seconds while the range is from 1 through 10 seconds.

8. Enable the error disable timeout timer.

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

   This parameter enables a timer that brings a port out of the disabled state. By default, the timeout feature is disabled.

9. Set the error disable timeout timer.

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

   When enabled the default is 300 seconds and the range is from 10 through 1000000 seconds.

10. Configure the port channel path cost.

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

    Specifying **custom** means the path cost changes according to the port channel's bandwidth.

11. Return to privileged EXEC mode.

```
device(config-stp)# end
```

12. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: Spanning Tree Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 -------------------------------------------------------------
 Eth 0/2      DES   FWD  2000       128   P2P            No
 Eth 0/20     DIS   DIS  20000000   128   P2P            No
 Eth 0/25     DIS   DIS  20000000   128   P2P            No
 Eth 0/30     DIS   DIS  20000000   128   P2P            No
```

```
 Eth 0/31     DIS   DIS  2000000    128   P2P           No
```

Observe that the settings comply with the formula set out in the STP parameter configuration section, as:

```
(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))
```

Or in this case `38 ≥ 25 ≥ 18`.

13. Save the configuration.

```
device# copy running-config startup-config
```

STP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# description stpForInterface
device(config-stp)# bridge-priority 4096
device(config-stp)# forward-delay 20
device(config-stp)# max-age 25
device(config-stp)# hello-time 8
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config-stp)# port-channel path-cost custom
device(config-stp)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Enabling and configuring STP on an interface

Globally enable STP and STP parameters.

The parameters can be configured individually by:

1. Entering the commands in steps 1-3
2. Running the relevant parameter command
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 0/20
```

3. Enable the interface.

```
device(conf-if-eth-0/20)# no shutdown
```

4. Configure the path cost for spanning tree calculations on the interface.

```
device(conf-if-eth-0/20)# spanning-tree cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.
5. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/20)# spanning-tree port-fast bpdu-guard
```

   BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.
6. Configure Root Guard on the interface.

```
device(conf-if-eth-0/20)# spanning-tree guard root
```

   Root Guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.
7. Specify an interface link-type.

```
device(conf-if-eth-0/20)# spanning-tree link-type point-to-point
```

   Specifying a point-to-point link enables rapid spanning tree transitions to the forwarding state. Specifying a shared link disables spanning tree rapid transitions. The default setting is point-to-point.
8. Specify port priority to influence the selection of root or designated ports.

```
device(conf-if-eth-0/20)# spanning-tree priority 64
```

   The range is from 0 through 240 in increments of 16. The default value is 128.
9. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: Spanning Tree Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 -------------------------------------------------------------
 Eth 0/2      DES   FWD  2000       128   P2P            No
 Eth 0/20     DES   FWD  1000        64   P2P            No
 Eth 0/25     DIS   DIS  20000000   128   P2P            No
 Eth 0/30     DIS   DIS  20000000   128   P2P            No
```

```
 Eth 0/31     DIS   DIS  2000000    128   P2P           No
```

Note

Observe that the settings comply with the formula set out in the STP parameters section, as:

`(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))`

or in this case: `38 ≥ 25 ≥ 18`

```
device# show running-config interface ethernet 0/20
interface ethernet 0/20
switchport
switchport mode access
switchport access val 1
spanning-tree cost 1000
spanning-tree guard root
spanning-tree link-type point-to-point
spanning-tree portfast bpdu-guard
spanning-tree priority 64
```

10. Save the settings by copying the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

STP on an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/20
device(conf-if-eth-0/20)# no shutdown
device(conf-if-eth-0/20)# spanning-tree cost 10000
device(conf-if-eth-0/20)# spanning-tree port-fast bpdu-guard
device(conf-if-eth-0/20)# spanning-tree guard root
device(conf-if-eth-0/20)# spanning-tree link-type point-to-point
device(conf-if-eth-0/20)# spanning-tree priority 64
device(conf-if-eth-0/20)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Configuring basic STP parameters

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable STP globally

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

3. Name the STP.

```
device(config-stp)# description stp1
```

4. Designate the root switch.

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

   The priority values can be set only in increments of 4096. The range is 0 through 61440.
5. Specify the bridge forward delay.

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

6. Configure the maximum aging time.

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

7. Configure the maximum hello time.

```
device(config-stp)# hello-time 8
```

8. Enable the error disable timeout timer.

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

9. Set the error disable timeout timer interval.

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

10. Enable port fast on switch ports.
    a. Configure port fast on Ethernet port 0/1.

```
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree portfast
device(conf-if-eth-0/1)# exit
```

       Spanning trees are automatically enabled on switch ports.
    b. Configure port fast on Ethernet port 0/2.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree portfast
device(conf-if-eth-0/2)# exit
```

    c. Repeat these commands for every port connected to workstations or PCs.

```
device(config)# interface ethernet ...
```

11. Specify port priorities to influence the selection of the root and designated ports.

```
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree priority 1
device(conf-if-eth-0/1)# exit
```

12. Enable the guard root feature.

```
device(config)# interface ethernet 0/12
device(conf-if-eth-0/12)# no shutdown
device(conf-if-eth-0/12)# spanning-tree guard root
```

    Root guard lets the device top participate in the STP but only when the device does not attempt to become the root.

13. Return to privileged exec mode.

```
device(conf-if-eth-0/12)# end
```

14. Verify the configuration.

```
device# show spanning-tree brief
Spanning-tree Mode: Spanning Tree Protocol
Root ID Priority 4096
Address 768e.f805.5800
Hello Time 8, Max Age 25, Forward Delay 20
Bridge ID Priority 4096
Address 768e.f805.5800
Hello Time 8, Max Age 25, Forward Delay 20
Interface     Role     Sts     Cost     Prio     Link-type     Edge
-----------------------------------------------------------------------
Eth 0/1       DES      FWD     2000     128      P2P           No
Eth 0/2       DES      FWD     2000     128      P2P           No
Eth 0/12      DES      FWD     2000     128      P2P           No
```

Observe that the settings comply with the formula set out in the STP parameter configuration section, as:

`(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))`

or in this case `38 ≥ 25 ≥ 18`.

15. Save the configuration.

```
device# copy running-config startup-config
```

Basic STP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# description stp1
device(conf-stp)# bridge-priority 28672
device(config-stp)# forward-delay 20
device(config-stp)# max-age 25
device(config-stp)# hello-time 8
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree portfast
device(conf-if-eth-0/1)# exit
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree portfast
device(conf-if-eth-0/2)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree priority 1
device(conf-if-eth-0/1)# exit
device(config)# interface ethernet 0/12
device(conf-if-eth-0/12)# no shutdown
device(conf-if-eth-0/12)# spanning-tree guard root
device(conf-if-eth-0/12)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

Re-enabling an error-disabled port automatically

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter STP configuration mode.

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

3. Enable the error-disable-timeout timer.

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

4. Set an interval after which port shall be enabled.

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

   The interval range is from 0 to 1000000 seconds, the default is 300 seconds.

5. Return to privileged EXEC mode.

```
device(config-stp)# end
```

6. Verify the configuration.

```
device# show spanning-tree
 Spanning-tree Mode: Spanning Tree Protocol

 Root Id: 8000.768e.f805.5800 (self)
 Bridge Id: 8000.768e.f805.5800

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: enabled
 Bpdu-guard errdisable timeout interval: 60 sec
```

Automatically re-enable an error-disabled port configuration example

```
device# configure terminal
device(config)# protocol spanning-tree stp
device(config-stp)# error-disable-timeout enable
device(config-stp)# error-disable-timeout interval 60
device(config-stp)# end
device# show spanning-tree
```

Clearing spanning tree counters

1. Clear spanning tree counters on all interfaces.

```
device# clear spanning-tree counter
```

2. Clear spanning tree counters on a specified Ethernet interface.

```
device# clear spanning-tree counter interface ethernet 0/3
```

3. Clear spanning tree counters on a specified port channel interface.

```
device# clear spanning-tree counter interface port-channel 12
```

Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

```
device# clear spanning-tree detected-protocols
```

2. Restart the spanning tree migration process on a specific Ethernet interface.

```
device# clear spanning-tree detected-protocols interface ethernet 0/3
```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

Port channel interface numbers range from 1 through 64.

Shutting down STP

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down STP.
   - Shut down STP globally and return to privileged EXEC mode.

```
device(config)# protocol spanning-tree stp
device(config-stp)# shutdown
device(config-stp)# end
```

   - Shut down STP on a specific interface and return to privileged EXEC mode.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree shutdown
device(conf-if-eth-0/2)# end
```

   - Shut down STP on a specific VLAN and return to privileged EXEC mode.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
```

3. Verify the configuration.

```
device# show spanning-tree
device#
```

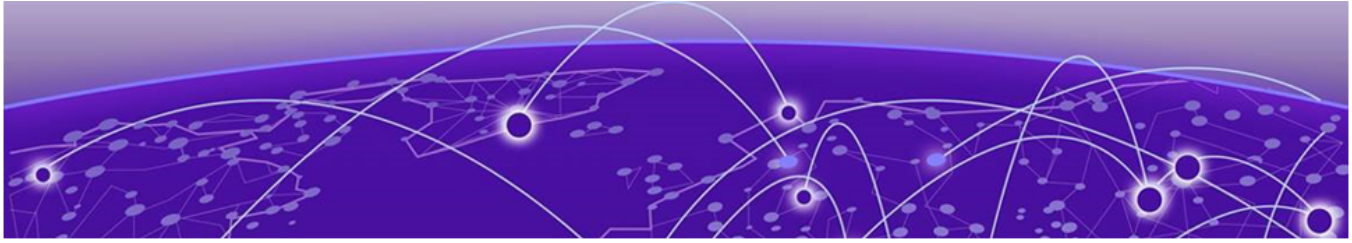
4. Save the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

Shut down STP configuration example

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-stp)# end
device# show spanning-tree
device# copy running-config startup-config
```

Note
Shutting down STP on a VLAN is used in this example.



802.1w Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol overview on page 205
Configuring RSTP on page 207

Rapid Spanning Tree Protocol overview

The STP (802.1d) standard was designed at a time when recovering connectivity after an outage within a minute or so was considered adequate performance. With the advent of Layer 3 switching in LAN environments, bridging competes with routed solutions where protocols such as OSPF are able to provide an alternate path in less time.

RSTP can be seen as evolution of the STP standard. It provides rapid convergence of connectivity following the failure of bridge, a bridge port, or a LAN. It provides rapid convergence of edge ports, new root ports, and ports connected through point-to-point links. The port, which qualifies for fast convergence, is derived from the duplex mode of a port. A port operating in full-duplex will be assumed to be point-to-point, while a half-duplex port will be considered as a shared port by default. This automatic setting can be overridden by explicit configuration.

RSTP is designed to be compatible and interoperate with STP. However, the benefit of RSTP fast convergence is lost when interacting with legacy STP (802.1d) bridges since RSTP downgrades itself to STP when it detects a connection to a legacy bridge.

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

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

The RSTP port roles for the interface are also different. 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 in 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. When the network is stable, the root and designated ports are in the forwarding state, while the alternate and backup ports are in the discarding state. When there is a topology change, the new RSTP port roles allow a faster transition of an alternate port into the forwarding state.

For more information about spanning trees, see the introductory sections in #unique_220.

RSTP Parameters

The parameters you would normally set when configuring STP are applicable to RSTP. Before you configure RSTP see the STP parameters sections for descriptions of the bridge parameters, the error disable timeout parameter, and the port channel path cost parameter.

There is one parameter that can be configured in RSTP that is not available in STP; the transmit hold count. This parameter configures the BPDU burst size by specifying the maximum number of BPDUs transmitted per second for before pausing for 1 second. The range is 1 through 10 while the default is 6. See Configuring RSTP on page 207 for the procedures to configure this parameter.

The edge port and auto edge features can be enabled in RSTP as well. See the section Edge port and automatic edge detection on page 206 and the section Enabling and configuring RSTP on an interface on page 209 for descriptions of these features and how they are configured.

Edge port and automatic edge detection

From an interface, you can configure a device to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

Follow these guidelines to configure a port as an edge port:

- When edge port is enabled, the port still participates in a spanning tree.
- A port can become an edge port if no BPDU is received.

- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

Note
If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

Configuring RSTP

Enabling and configuring RSTP globally

See the section STP parameters for parameters applicable to all STP variants.

You can enable RSTP or RSTP with one or more parameters enabled. The parameters can be enabled or changed individually by entering the commands in steps 1 and 2, running the parameter command, verifying the result, and then saving the configuration.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable RSTP.

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

   You can shut down RSTP by entering the **shutdown** command.

3. Designate the root device.

```
device(conf-rstp)# bridge-priority 28582
```

   The range is 0 through 61440 and the priority values can be set only in increments of 4096.

   You can shut down RSTP by entering the **shutdown** command when in RSTP configuration mode.

4. Configure the bridge forward delay value.

```
device(conf-rstp)# forward-delay 15
```

5. Configure the bridge maximum aging time value.

```
device(conf-rstp)# max-age 20
```

6. Enable the error disable timeout timer.
   a. Enable the timer.

```
device(conf-rstp)# error-disable-timeout enable
```

b. Configure the error disable timeout interval value.

```
device(conf-rstp)# error-disable-timeout interval 60
```

7. Configure the port-channel path cost.

```
device(conf-rstp)# port-channel path-cost custom
```

8. Configure the bridge hello-time value.

```
device(conf-rstp)# hello-time 2
```

9. Specify the transmit hold count.

```
device(config-rstp)# transmit-holdcount 5
```

This command configures the maximum number of BPDUs transmitted per second.

10. Return to privileged exec mode.

```
device(conf-rstp)# end
```

11. Verify the configuration

```
device# show spanning-tree

 Spanning-tree Mode: Rapid Spanning Tree Protocol

 Root Id: 8000.01e0.5200.0180 (self)
 Bridge Id: 8000.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: enabled
 Bpdu-guard errdisable timeout interval: 60 sec
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

12. Save the configuration.

```
device# copy running-config startup-config
```

Enabling RSTP and configuring RSTP parameters example

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(conf-rstp)# bridge-priority 28582
device(conf-rstp)# forward-delay 20
device(conf-rstp)# max-age 25
device(conf-rstp)# error-disable-timeout enable
device(conf-rstp)# error-disable-timeout interval 60
device(conf-rstp)# port-channel path-cost custom
device(conf-rstp)# hello-time 5 forward-delay 16 max-age 21
```

```
device(conf-rstp)# transmit-holdcount 5
device(conf-rstp# end
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring RSTP on an interface

You can configure the parameters individually on an interface by doing the following:

1. Entering the commands in Steps 1 through 3.
2. Specifying additional parameters, as appropriate.
3. Verifying the result.
4. Saving the configuration.

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface subtype configuration mode.

   ```
   device(config)# interface ethernet 0/10
   ```

3. Enable the interface.

   ```
   device(conf-if-eth-0/10)# no shutdown
   ```

   To disable the spanning tree on the interface you use the **spanning-tree shutdown** command.

4. Specify the port priority on the interface.

   ```
   device(conf-if-eth-0/10)# spanning-tree priority 128
   ```

   The range is from 0 through 240 in increments of 16. The default value is 128.

5. Specify the path cost on the interface.

   ```
   device(conf-if-eth-0/10)# spanning-tree cost 20000000
   ```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

6. Enable edge port.

   ```
   device(conf-if-eth-0/10)# spanning-tree edgeport
   ```

   If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

7. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/10)# spanning-tree edgeport bpdu-guard
```

   BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.
8. Enable automatic edge detection on the interface.

```
device(conf-if-eth-0/10)# spanning-tree autoedge
```

   You use this command to automatically identify the edge port. A port becomes an edge port if it receives no BPDUs. By default, automatic edge detection is disabled.
9. Enable root guard on the interface.

```
device(conf-if-eth-0/10)# spanning-tree guard root
```

   Root guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.
10. Specify a link type on the interface.

```
device(conf-if-eth-0/10)# spanning-tree link-type point-to-point
```

Note
The link type is explicitly configured as **`point-to-point`** rather than **`shared`**.

11. Return to privileged EXEC mode.

```
device(conf-if-eth-0/10)# end
```

12. Verify the configuration.

```
device# show spanning-tree

 Spanning-tree Mode: Rapid Spanning Tree Protocol

 Root Id: 8000.01e0.5200.0180 (self)
 Bridge Id: 8000.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: enabled
 Bpdu-guard errdisable timeout interval: 60 sec


 Port Eth 0/10 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Spanning Tree Protocol - Received None - Sent STP
```

```
    Edgeport: on; AutoEdge: yes; AdminEdge: no; EdgeDelay: 3 sec
    Configured Root guard: on; Operational Root guard: on
    Bpdu-guard: on
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

Interface   Role     Sts     Cost       Prio     Link-type     Edge
--------------------------------------------------------------------
Eth 0/10    DES      FWD     20000000   128      P2P           No
```

The **forward-delay**, **hello-time**, and **max-age** parameters are set globally, not on the interface.

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

13. Save the configuration.

```
device# copy running-config startup-config
```

RSTP on an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/10
device(conf-if-eth-0/10)# no spanning-tree shutdown
device(conf-if-eth-0/10)# spanning-tree priority 128
device(conf-if-eth-0/10)# spanning-tree cost 20000000
device(conf-if-eth-0/10)# spanning-tree edgeport
device(conf-if-eth-0/10)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-0/10)# spanning-tree autoedge
device(conf-if-eth-0/10)# spanning-tree guard root
device(conf-if-eth-0/10)# spanning-tree link-type point-to-point
device(conf-if-eth-0/10)# end
device# show spanning-tree
device# copy running-config startup-config
```

Configuring basic RSTP parameters

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable RSTP.

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

3. Designate the root device.

```
device(conf-rstp)# bridge-priority 28582
```

4. Enable the error disable timeout timer value.

```
device(conf-rstp)# error-disable-timeout enable
```

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

```
device(conf-rstp)# error-disable-timeout interval 60
```

6. Enable edge port on switch ports.
   a. Enter interface subtype configuration mode for the switchport.

```
device(conf-rstp)# interface ethernet 0/10
```

   b. Enable edge port.

```
device(conf-if-eth-0/10)# spanning-tree edge-port
```

   c. Return to global configuration mode.

```
device(conf-if-eth-0/10)# exit
```

   d. Repeat the above steps for all ports that connect to a workstation or PC.
7. Specify port priorities.
   a. Enter interface subtype configuration mode.

```
device(config)# interface ethernet 0/11
```

   b. Configure the port priority.

```
device(conf-if-eth-0/11)# spanning-tree priority 1
```

   c. Return to global configuration mode.

```
device(conf-if-eth-0/11)# exit
```

8. Enable the guard root feature.
   a. Enter interface configuration mode.

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

   b. Configure the port priority.

```
device(conf-if-eth-0/1)# spanning-tree guard root
```

   c. Return to privileged EXEC mode.

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

9. Verify the configuration.

```
device# show spanning-tree

Spanning-tree Mode: Rapid Spanning Tree Protocol

Root Id: 4096.01e0.5200.0180 (self)
Bridge Id: 4096.01e0.5200.0180
Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
Number of topology change(s): 0
Bpdu-guard errdisable timeout: disabled
Bpdu-guard errdisable timeout interval: 300 sec
switch# show spanning-tree brief
Spanning-tree Mode: Rapid Spanning Tree Protocol
```

```
Root ID Priority 4096
Address 768e.f805.5800
Hello Time 2, Max Age 20, Forward Delay 15
Bridge ID Priority 4096
Address 768e.f805.5800

Interface   Role     Sts     Cost     Prio     Link-type     Edge
-------------------------------------------------------------------
Eth 0/1     DES      FWD     2000     128      P2P           No
Eth 0/10    DES      FWD     2000     128      P2P           No
Eth 0/11    DES      FWD     2000     128      P2P           No
```

Observe that the settings comply with the formula set out in the STP parameters section, as follows:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

10. Save the configuration.

```
device# copy running-config startup-config
```

Basic RSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rstp
device(conf-rstp)# bridge-priority 28582
device(conf-rstp)# error-disable-timeout enable
device(conf-rstp)# error-disable-timeout interval 60
device(conf-rstp)# interface etherne 0/10
device(conf-if-eth-0/10)# spanning-tree edge-port
device(conf-if-eth-0/10)# exit
device(config)# interface ethernet 0/11
device(conf-if-eth-0/11)# spanning-tree priority 1
device(conf-if-eth-0/11)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# spanning-tree guard root
device(conf-if-eth-0/1)# exit
device# show spanning-tree
device# copy running-config startup-config
```

Clearing spanning tree counters

1. Clear spanning tree counters on all interfaces.

```
device# clear spanning-tree counter
```

2. Clear spanning tree counters on a specified Ethernet interface.

```
device# clear spanning-tree counter interface ethernet 0/3
```

3. Clear spanning tree counters on a specified port channel interface.

```
device# clear spanning-tree counter interface port-channel 12
```

Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

```
device# clear spanning-tree detected-protocols
```

2. Restart the spanning tree migration process on a specific Ethernet interface.

```
device# clear spanning-tree detected-protocols interface ethernet 0/3
```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

Port channel interface numbers range from 1 through 64.

Shutting down RSTP

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down RSTP.
   - Shut down STP globally and return to privileged EXEC mode.

```
device(config)# protocol spanning-tree rstp
device(conf-rstp)# shutdown
device(conf-rstp)# end
```

   - Shut down RSTP on a specific interface and return to privileged EXEC mode.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree shutdown
device(conf-if-eth-0/2)# end
```

   - Shut down RSTP on a specific VLAN and return to privileged EXEC mode.

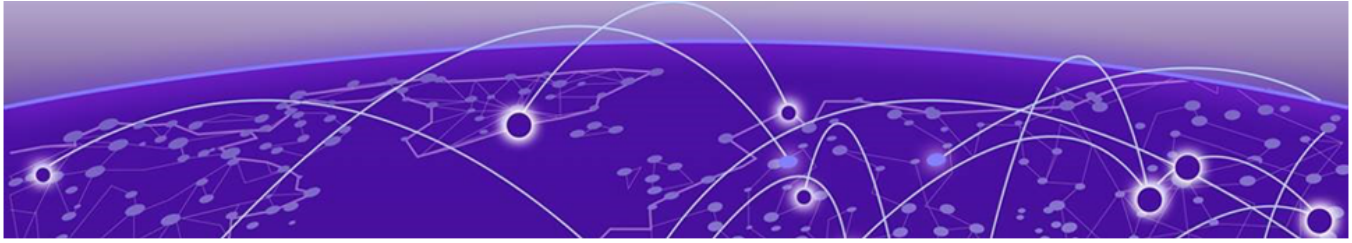
```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
```

3. Verify the configuration.

```
device# show spanning-tree
device#
```

4. Save the configuration.

```
device# copy running-config startup-config
```



Per-VLAN Spanning Tree+ and Rapid Per-VLAN Spanning Tree+

PVST+ and R-PVST+ overview on page 215
Configuring PVST+ and R-PVST+ on page 222

PVST+ and R-PVST+ overview

Both the STP and the RSTP build a single logical topology. A typical network has multiple VLANs. A single logical topology does not efficiently utilize the availability of redundant paths for multiple VLANs. A single logical topology does not efficiently utilize the availability of redundant paths for multiple VLANs. If a port is set to the blocked state or the discarding state for one VLAN (under the STP or the RSTP), it is the same for all other VLANs. PVST+ builds on the STP on each VLAN, and R-PVST+ builds on the RSTP on each VLAN.

PVST+ R-PVST+ provide interoperability with Cisco PVST and R-PVST and other vendor switches which implement Cisco PVST or R-PVST. the PVST+ and R-PVST+ implementations are extensions to PVST and R-PVST, which can interoperate with an STP topology, including MSTP (CIST), on Extreme and other vendor devices sending untagged IEEE BPDUs.

PVST+ and R-PVST+ guidelines and restrictions

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

- Extreme supports PVST+ and R-PVST+ only. The PVST and R-PVST protocols are proprietary to Cisco and are not supported.
- A port native VLAN is the native VLAN ID associated with a trunk port on an Extreme switch. This VLAN ID is associated with all untagged packets on the port. The default native VLAN ID for a trunk port is 1.
- IEEE compliant switches run just one instance of STP protocol shared by all VLANs, creating a Mono Spanning Tree (MST). A group of such switches running a single spanning tree forms an MST region.
- You can configure up to 256 PVST+ or R-PVST+ instances. If you have more than 256 VLANs configured on the switch and enable PVST then the first 256 VLANs are PVST/+ or R-PVST+ enabled.

- In PVST/+ or R-PVST+ mode, when you are connected to a Cisco or MLX switch, the Cisco proprietary MAC address to which the BPDUs are sent/processed must be explicitly configured on a per-port basis.
- In PVST/+ or R-PVST+ mode, when you connect to a Cisco switch using a trunk port, the Extreme switch must have a native VLAN configured on the trunk port (same configuration as on the other side).
- A Common Spanning Tree (CST) is the single spanning tree instance used by Extreme switches to interoperate with 802.1q bridges. This spanning tree instance stretches across the entire network domain (including PVST, PVST+ and 802.1q regions). It is associated with VLAN 1 on the Extreme switch.
- In order to interact with STP and IEEE 802.1q trunk, PVST evolved to PVST+ to interoperate with STP topology by STP BPDU on the native or default VLAN.
- A group of switches running PVST+ is called a PVST+ region.

For more information about spanning trees, see the introductory sections in the Spanning Tree Protocol chapter.

PVST+ and R-PVST+ parameters

The parameters you would normally set when you configure STP are applicable to PVST+ and R-PVST+. Before you configure PVST+ or R-PVST+ parameters see the sections in the Standing Tree Protocol chapter explaining bridge parameters, the error disable timeout parameter and the port channel path cost parameter.

There is one parameter that can be configured in R-PVST+ that is not available in STP or PVST+; the transmit hold count. This parameter configures the BPDU burst size by specifying the maximum number of BPDUs transmitted per second for before pausing for 1 second. The range is 1 through 10 while the default is 6. See the section Configuring R-PVST+ for the procedure to configure this parameter.

Bridge protocol data units in different VLANs

Across IEEE 802.1q trunks, Extreme switches run PVST+. The goal is to interoperate with standard IEEE STP (or RSTP or MSTP), while transparently tunneling PVST+ instance BPDUs across the MST region to potentially connect to other Extreme switches across the MST region.

On trunk ports that allow VLAN 1, PVST+ also sends PVST+ BPDUs to a Cisco-proprietary multicast MAC address (0100.0ccc.cccd) or Extreme-proprietary multicast MAC address (0304.0800.0700) depending on the configuration. By default, the PVST+ BPDUs are sent to Extreme-proprietary multicast MAC address on Extreme switches. These BPDUs are tunneled across an MST region. The PVST+ BPDUs for VLAN 1 are only used for the purpose of consistency checks and that it is only the IEEE BPDUs that are used for building the VLAN 1 spanning tree. So in order to connect to the CST, it is necessary to allow VLAN 1 on all trunk ports.

For all other VLANs, PVST+ BPDUs are sent on a per-VLAN basis on the trunk ports. These BPDUs are tunneled across an MST region. Consequently, for all other VLANs, MST region appears as a logical hub. The spanning tree instances for each VLAN in one

PVST+ region map directly to the corresponding instances in another PVST+ region and the spanning trees are calculated using the per-VLAN PVST+ BPDUs.

Similarly, when a PVST+ region connects to a MSTP region, from the point of view of MSTP region, the boundary bridge thinks it is connected to a standard IEEE compliant bridge sending STP BPDUs. So it joins the CIST of the MSTP region to the CST of the PVST+ region (corresponding to VLAN 1). The PVST+ BPDUs are tunneled transparently through the MSTP region. So from the Extreme bridge point of view, the MSTP region looks like a virtual hub for all VLANs except VLAN 1.

The PVST+ BPDUs are sent untagged for the native VLAN and tagged for all other VLANs on the trunk port.

On access ports, Extreme switches run classic version of IEEE STP/RSTP protocol, where the BPDUs are sent to the standard IEEE multicast address "0180.C200.0000". So if we connect a standard IEEE switch to an access port on the Extreme switch, the spanning tree instance (corresponding to the access VLAN on that port) of the Extreme switch is joined with the IEEE STP instance on the adjacent switch.

For introductory information about STP BPDUs, see the section BPDUs on page 188.

BPDU configuration notes

BPDUs are sent to a Cisco-proprietary multicast MAC address 0100.0ccc.cccd or Extreme-proprietary multicast MAC address 0304.0800.0700.By default, the PVST+ BPDUs are sent to Extreme-proprietary multicast MAC address on Extreme switches. These are called SSTP (Single Spanning Tree Protocol) BPDUs. The format of the SSTP BPDU is nearly identical to the 802.1d BPDU after the SNAP header, except that a type-length-value (TLV) field is added at the end of the BPDU. The TLV has 2 bytes for type (0x0), 2 bytes for length, and 2 bytes for the VLAN ID. See Table 39 on page 218 and Table 40 on page 218 for an outline of the BPDU header content.

Topology Change Notification (TCN) BPDUs are used to inform other switches of port changes. TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

In PVST+, three types of TCN BPDUs are sent out depending on the type of the link. See Table 42 on page 220 and Table 43 on page 221.

- Standard IEEE TCN BPDU.
- Untagged TCN BPDU sent to the Cisco/Extreme proprietary MAC address.
- Tagged TCN BPDU sent to the Cisco/Extreme proprietary MAC address.

BPDU R-PVST+ header and field comparisons

Table 39: Extreme R-PVST+ BPDU headers/fields

Header/field	Standard IEEE STP/ RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
Source Address (MAC SA)	6B	6B	6B
Destination Address (MAC DA)	0180C2.000000 (6B)	030408.000700 (6B)	030408.000700 (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
Source Service Access Point (SSAP)	42	AA 03	AA 03
Destination Service Access Point (DSAP)	42	AA	AA
Extreme Organizationally Unique Identifier (OUI)	-	02 04 08	02 04 08
PVST PID	-	01 0B	01 0B
Logical Link Control (LLC)	3B	+	+
SubNetwork Access Protocol (SNAP)	-	Yes (2B)	Yes (2B)
IEEE BPDU INFO	35B	35B	35B
Type, Length, Value (TLV) Pad Type Length VLAN ID	-	6B 00 (1B) 00 00 00 02 2B	6B 00 (1B) 00 00 00 02 2B

Table 40: Cisco R-PVST+ BPDU headers/fields

Header/field	Standard IEEE STP/ RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	01000C.CCCCCD (6B)	010002.CCCCCD (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B

Table 40: Cisco R-PVST+ BPDU headers/fields (continued)

Header/field	Standard IEEE STP/ RSTP BPDU (64B padded)	R-PVST+ untagged BPDU (64B padded)	R-PVST+ tagged BPDU (72B padded)
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	00 00 0C	00 00 0C
PVST PID	-	01 0B	01 0B
LLC	3B	+	+
SNAP	-	Yes	Yes
IEEE BPDU INFO	35B	35B	35B
TLV Pad Type Length VLAN ID	-	6B 00 (1B) 00 00 00 02 2B	6B 00 (1B) 00 00 00 02 2B

Sent BPDUs

1. For all tagged VLANs on the port on which PVST+ is enabled, 802.1q tagged SSTP BPDUs are sent to the Cisco or Extreme MAC address. The 802.1q tag contains the VLAN ID. (VLAN 1 could be tagged on the port. In that case a tagged BPDU for VLAN 1 is sent). The IEEE compliant switches do not consider these BPDUs as a control packet. So they forward the frame as they would forward to any unknown multicast address on the specific VLAN.
2. If PVST+ is enabled on the untagged (native) VLAN of the port, an untagged SSTP BPDU is sent to the Extreme or Cisco MAC address on the native VLAN of the trunk. It is possible that the native VLAN on the Extreme or Cisco port is not VLAN 1. This BPDU is also forwarded on the native VLAN of the IEEE 802.1q switch just like any other frame sent to an unknown multicast address.
3. In addition to the above SSTP BPDUs, a standard IEEE BPDU (802.1d) is also sent, corresponding to the information of VLAN 1 on the Extreme or Cisco switch. This BPDU is not sent if VLAN 1 is explicitly disabled on the trunk port.

The following table lists the types of BPDUs sent in case of different port types. The numbers in the third column are the VLAN instance for which these BPDUs are sent/ processed.

Table 41: Types of BPDUs sent for different port types

Port Configuration	Extreme or Cisco - PVST(+)	VLAN instance
Access - VLAN 1	Standard IEEE BPDU (64B)	1
Access - VLAN 100	Standard IEEE BPDU (64B)	100

Table 41: Types of BPDUs sent for different port types (continued)

Port Configuration	Extreme or Cisco - PVST(+)	VLAN instance
Trunk - Native VLAN 1 Allowed VLANs - 1, 100, 200	Standard IEEE BPDU (64B) Extreme or Cisco untagged BPDU (68B) Extreme or Cisco tagged BPDU (72B) Extreme or Cisco tagged BPDU (72B)	1 1 100 200
Trunk - Native VLAN 100 Allowed VLANs - 1, 100, 200	Standard IEEE BPDU (64B) Extreme or Cisco untagged BPDU (68B) Extreme or Cisco tagged BPDU (72B Extreme or Cisco tagged BPDU (72B)	1 100 1 200
Trunk - Native VLAN 100 Allowed VLANs - 100	Extreme or Cisco untagged BPDU (68B)	100
Trunk - Native VLAN 100 Allowed VLANs - 100, 200	Extreme or Cisco untagged BPDU (68B) Extreme or Cisco tagged BPDU (72B)	100 200

TCN headers and fields

For introductory information about STP BPDUs, see the section TCN BPDUs on page 189.

Table 42: Extreme PVST+ TCN BPDU headers/fields

Header/field	Standard IEEE STP TCN BPDU (64B with padding)	PVST+ untagged TCN BPDU (64B with padding)	PVST+ tagged TCN BPDU (68B with padding)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	030408.000700 (6B)	030408.000700 ((6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	02 04 08	02 04 08
PVST PID	-	01 0B	01 0B
LLC	3B	8B	8B
SNAP	4B	Entire BPDU with type = TCN 35B	Entire BPDU with type = TCN 35B

Table 43: Cisco PVST TCN BPDU headers/fields

Header/field	Standard IEEE STP TCN BPDU (64B padded)	PVST untagged TCN BPDU (64B padded)	PVST tagged TCN BPDU (68B padded)
MAC SA	6B	6B	6B
MAC DA	0180C2.000000 (6B)	01000C.CCCCCD (6B)	01000C.CCCCCD (6B)
Length	2B	2B	-
Type	-	-	81 00 (2B)
802.1q tag	-	-	4B
SSAP	42 03	AA 03	AA 03
DSAP	42	AA	AA
Cisco OUI	-	00 00 0C	00 00 0C
PVST PID	-	01 0B	01 0B
LLC	3B	8B	8B
SNAP	-	Yes	Yes
IEEE TCN BPDU INFO	4B	Entire BPDU with type = TCN 35B	Entire BPDU with type = TCN 35B

PortFast

Consider the following when configuring PortFast:

- Do not enable PortFast on ports that connect to other devices.
- PortFast only needs to be enabled on ports that connect to workstations or PCs. Repeat this configuration for every port connected to workstations or PCs.
- Enabling PortFast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.
- If BPDUs are received on a PortFast- enabled interface, the interface loses the edge port status unless it receives a **`shutdown`**/**`no shutdown`** command.
- PortFast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

Edge port and automatic edge detection

From an interface, you can configure a device to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

Follow these guidelines to configure a port as an edge port:

- When edge port is enabled, the port still participates in a spanning tree.
- A port can become an edge port if no BPDU is received.

- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

Note
If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

Configuring PVST+ and R-PVST+

Enabling and configuring PVST+ globally

You can enable PVST+ with one or more parameters configured. The parameters can be configured or changed individually by entering the commands in steps 1 and 2, running the parameter command, verifying the result, and then saving the configuration.

For more information about spanning trees and spanning tree parameters, see the introductory sections in the Spanning Tree Protocol chapter.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable PVST+.

```
device(config)# protocol spanning-tree pvst
```

3. Configure the bridge priority for the common instance.

```
device(config-pvst)# bridge-priority 4096
```

   Valid values range from 0 through 61440 in increments of 4096. Assigning a lower priority value indicates that the bridge might become root.

   You can shut down PVST+ by entering the **shutdown** command when in PVST configuration mode.

4. Configure the forward delay parameter.

```
device(config-pvst)# forward-delay  11
```

5. Configure the hello time parameter.

```
device(config-pvst)# hello-time 2
```

6. Configure the maximum age parameter.

```
device(config-pvst)# max-age 7
```

7. Return to privileged exec mode.

```
device(config-pvst)# end
```

8. Verify the configuration.

```
device#  show spanning-tree brief
VLAN 1

 Spanning-tree Mode: PVST Protocol

      Root ID      Priority 4097
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 4097
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------

VLAN 100

 Spanning-tree Mode: PVST Protocol

      Root ID      Priority 4196
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 4196
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------
```

   Observe that the settings comply with the formula set out in the STP parameters section, as:

   (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

   or in this case: 20 ≥7 ≥ 6.

9. Save the configuration.

```
device# copy running-config startup-config
```

PVST+ configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(config-pvst)# bridge-priority 4096
device(config-pvst)# forward-delay 11
device(config-pvst)# hello-time 2
device(config-pvst)# max-age 7
device(config-pvst)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

For more information about configuring PVST+ parameters, see STP parameters on page 192. PVST+, R-PVST+, and other types of spanning trees share many tasks with STP.

Enabling and configuring PVST+ on an interface

The ports and parameters can be configured individually on a system by:

1. Entering the commands in steps 1, and 2
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable PVST+.

```
device(config)# protocol spanning-tree pvst
```

3. Enter interface configuration mode.

```
device(config-pvst)# interface ethernet 0/3
```

4. Enable spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

5. Configure the interface link type.

```
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
```

6. Specify the port priority to influence the selection of root or designated ports.

```
device(conf-if-eth-0/3)# spanning-tree priority 64
```

   The range is from 0 through 240 in increments of 16. The default value is 128.

7. Configure the path cost for spanning tree calculations on the interface.

```
device(conf-if-eth-0/3)# spanning-tree cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

8. Configure the path cost for spanning tree calculations a specific VLAN.

```
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.

9. Enable root guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree guard root
```

   Root guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.

10. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree port-fast bpdu-guard
```

    BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

11. Enable BPDU filtering on the interface.

```
device(conf-if-eth-0/3)# spanning-tree port-fast bpdu-filter
```

    BPDU filtering allows you to avoid transmitting BPDUs on ports that are connected to an end system.

12. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

13. Verify the configuration.

```
device# show spanning-tree brief

 Spanning-tree Mode: PVST Protocol

      Root ID      Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

      Bridge ID    Priority 4096
                   Address 768e.f805.5800
                   Hello Time 8, Max Age 25, Forward Delay 20

 Interface    Role  Sts  Cost       Prio  Link-type     Edge
 -------------------------------------------------------------
 Eth 0/3      DES   FWD  200000     64    P2P            No
```

    Observe that the settings comply with the formula set out in the STP parameters section, as:

    (2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

    or in this case :38 ≥ 25 ≥ 18.

14. Save the configuration.

```
device# copy running-config startup-config
```

PVST+ on an interface configuration example

```
device# configure terminal
```

```
device(config)# protocol spanning-tree pvst
device(conf-pvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
device(conf-if-eth-0/3)# spanning-tree priority 64
device(conf-if-eth-0/3)# spanning-tree cost 10000
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
device(conf-if-eth-0/3)# spanning-tree guard root
device(conf-if-eth-0/3)# spanning-tree port-fast bpdu-guard
device(conf-if-eth-0/3)# exit
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring PVST+ on a system

The ports and parameters can be configured individually on a system by:

1. Entering the commands in steps 1, and 2
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable PVST+.

   ```
   device(config)# protocol spanning-tree pvst
   ```

3. Configure the bridge priority for the common instance.

   ```
   device(config-pvst)# bridge-priority 4096
   ```

   Valid values range from 0 through 61440 in multiples of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the forward delay parameter.

   ```
   device(config-pvst)# forward-delay  15
   ```

5. Configure the hello time parameter.

   ```
   device(config-pvst)# hello-time 2
   ```

6. Configure the maximum age parameter.

   ```
   device(config-pvst)# max-age 20
   ```

7. Add VLANs.
   a. Configure VLAN 100 with a priority of 0.

      ```
      device(config-pvst)# vlan 100 priority 0
      ```

      The bridge priority in configured in multiples of 4096.

b. Configure VLAN 201 with a priority of 12288.

```
device(config-pvst)# vlan 201 priority 12288
```

c. Configure VLAN 301 with a priority of 20480.

```
device(config-pvst)# vlan 301 priority 20480
```

8. Set the switching characteristics for interface 0/3.
   a. Enter interface configuration mode.

```
device(config)# interface ethernet 0/3
```

   b. Set the switching characteristics of the interface.

```
device(conf-if-eth-0/3)# switchport
```

   c. Set the interface mode to trunk.

```
device(conf-if-eth-0/3)# switchport mode trunk
```

   d. Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
```

   e. Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
```

   f. Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
```

   g. Enable spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

   h. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

9. Set the switching characteristics for interface 0/4.
   a. Enter interface configuration mode.

```
device(config)# interface ethernet 0/4
```

   b. Set the switching characteristics of the interface.

```
device(conf-if-eth-0/4)# switchport
```

   c. Set the interface mode to trunk.

```
device(conf-if-eth-0/4)# switchport mode trunk
```

   d. Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
```

e. Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
```

f. Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
```

g. Enable spanning tree on the interface.

```
device(conf-if-eth-0/4)# no spanning-tree shutdown
```

h. Return to privileged EXEC mode.

```
device(conf-if-eth-0/4)# exit
```

10. To interoperate with switches other than VDX switches in PVST+ mode, you must configure the interface that is connected to that switch.
    a. Enter interface configuration mode for the port that interoperates with a VDX device.

```
device(config)# interface ethernet 0/12
```

    b. Specify the MAC address for the device.

```
device(conf-if-eth-0/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
```

    c. Enable spanning tree on the interface.

```
device(conf-if-eth-0/12)# no spanning-tree shutdown
```

    d. Return to privileged EXEC mode.

```
device(conf-if-eth-0/12)# end
```

11. Verify the configuration.

```
device# show spanning-tree

VLAN 1

 Spanning-tree Mode: PVST Protocol

 Root Id: 0001.01e0.5200.0180 (self)
 Bridge Id: 0001.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
```

```
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 100

 Spanning-tree Mode: PVST Protocol

 Root Id: 0064.01e0.5200.0180 (self)
 Bridge Id: 0064.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 201
```

```
 Spanning-tree Mode: PVST Protocol

 Root Id: 30c9.01e0.5200.0180 (self)
 Bridge Id: 30c9.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 301

 Spanning-tree Mode: PVST Protocol

 Root Id: 512d.01e0.5200.0180 (self)
 Bridge Id: 512d.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
```

```
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

12. Save the configuration.

```
device# copy running-config startup-config
```

Enable PVST+ on a system configuration example

```
device# configure terminal
device(config)# protocol spanning-tree pvst
device(config-pvst)# bridge-priority 4096
device(config-pvst)# forward-delay  15
device(config-pvst)# hello-time 2
device(config-pvst)# max-age 20
device(config-pvst)# vlan 100 priority 0
device(config-pvst)# vlan 201 priority 12288
device(config-pvst)# vlan 301 priority 20480
device(config-pvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# switchport
device(conf-if-eth-0/3)# switchport mode trunk
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# exit
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# switchport
device(conf-if-eth-0/4)# switchport mode trunk
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/4)# no spanning-tree shutdown
device(conf-if-eth-0/4)# end
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring R-PVST+ globally

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable R-PVST+.

```
device(config)# protocol spanning-tree rpvst
```

3. Configure the bridge priority for the common instance.

```
device(config-rpvst)# bridge-priority 4096
```

Valid priority values range from 0 through 61440 in multiples of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the forward delay parameter.

```
device(config-rpvst)# forward-delay  20
```

5. Configure the hello time parameter.

```
device(config-rpvst)# hello-time 22
```

6. Configure the maximum age parameter.

```
device(config-rpvst)# max-age 8
```

7. Set the transmit hold count for the bridge.

```
device(config-rpvst)# transmit-holdcount 9
```

This command configures the maximum number of BPDUs transmitted per second before pausing for 1 second. The range is 1 through 10. The default is 6.

8. Return to privileged exec mode.

```
device(config-rpvst)# end
```

9. Verify the configuration.

```
device# show spanning-tree brief
VLAN 1

 Spanning-tree Mode: Rapid PVST Protocol

      Root ID      Priority 4096
                   Address 01e0.5200.0180
                   Hello Time 2, Max Age 7, Forward Delay 11

      Bridge ID    Priority 32769
                   Address 01e0.5200.0180
                   Hello Time 8, Max Age 22, Forward Delay 20, Tx-HoldCount 9
                   Migrate Time 3 sec

 Interface     Role  Sts  Cost       Prio  Link-type        Edge
 ---------------------------------------------------------------------
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 20 ≥ 7 ≥ 6.

10. Save the configuration.

```
device# copy running-config startup-config
```

R-PVST+ configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# bridge-priority 4096
device(config-rpvst)# forward-delay 20
device(config-rpvst)# hello-time 22
device(config-rpvst)# max-age 8
device(config-rpvst)# transmit-holdcount 9
device(config-rpvst)# end
device# show spanning-tree brief
device# copy running-config startup-config
```

For more information about configuring parameters, see the section STP parameter configuration.

Enabling and configuring R-PVST+ on an interface

The ports and parameters can be configured individually on a system by:

1. Entering the commands in steps 1-3
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable R-PVST+.

```
device(config)# protocol spanning-tree rpvst
```

3. Enter interface configuration mode.

```
device(config-rpvst)# interface ethernet 0/3
```

4. Enable the spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

5. Configure the interface link type.

```
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
```

6. Specify the port priority to influence the selection of root or designated ports.

```
device(conf-if-eth-0/3)# spanning-tree priority 64
```

   The range of priority values is from 0 through 240 in multiples of 16. The default value is 128.
7. Configure the path cost for spanning tree calculations on the interface.

```
device(conf-if-eth-0/3)# spanning-tree cost 200000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.
8. Configure the path cost for spanning tree calculations a specific VLAN.

```
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
```

   The lower the path cost means a greater chance that the interface becomes the root port. The range is 1 through 200000000. The default path cost is assigned as per the port speed.
9. Enable automatic edge detection on the interface.

```
device(conf-if-eth-0/3)# spanning-tree autoedge
```

   You use this command to automatically identify the edge port. A port becomes an edge port if it receives no BPDUs. By default, automatic edge detection is disabled.
10. Enable root guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree guard root
```

    Root guard protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge.
11. Enable the spanning tree on the edge port.

```
device(conf-if-eth-0/3)# spanning-tree edgeport
```

    If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.
12. Enable BPDU guard on the interface.

```
device(conf-if-eth-0/3)# spanning-tree edgeport bpdu-guard
```

    BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.
13. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

14. Verify the configuration.

```
device# show spanning-tree brief
```

```
Spanning-tree Mode: Rapid PVST Protocol

     Root ID      Priority 4096
                  Address 768e.f805.5800
                  Hello Time 8, Max Age 25, Forward Delay 20

     Bridge ID    Priority 4096
                  Address 768e.f805.5800
                  Hello Time 8, Max Age 25, Forward Delay 20

Interface    Role  Sts  Cost       Prio  Link-type     Edge
-------------------------------------------------------------
Eth 0/3      DES   FWD  200000     128   P2P            No
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 38 ≥ 25 ≥ 18.

15. Save the configuration.

```
device# copy running-config startup-config
```

R-PVST+ on an interface configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# spanning-tree link-type point-to-point
device(conf-if-eth-0/3)# spanning-tree priority 64
device(conf-if-eth-0/3)# spanning-tree cost 200000
device(conf-if-eth-0/3)# spanning-tree vlan 10 cost 10000
device(conf-if-eth-0/3)# spanning-tree autoedge
device(conf-if-eth-0/3)# spanning-tree guard root
device(conf-if-eth-0/3)# spanning-tree edgeport
device(conf-if-eth-0/3)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-0/3)# exit
device# show spanning-tree
device# copy running-config startup-config
```

Enabling and configuring R-PVST+ on a system

The ports and parameters can be configured individually by:

1. Entering the commands in steps 1 and 2
2. Running the relevant addition steps and parameter commands
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable R-PVST+.

```
device(config)# protocol spanning-tree rpvst
```

   You can shut down R-PVST+ by entering the **shutdown** command when in `rpvst` configuration mode.

3. Configure the bridge priority for the common instance.

```
device(config-rpvst)# bridge-priority 4096
```

   Valid values range from 0 through 61440 in increments of 4096. Assigning a lower priority value indicates that the bridge might become root.

4. Configure the forward delay parameter.

```
device(config-rpvst)# forward-delay  20
```

5. Configure the hello time parameter.

```
device(config-rpvst)# hello-time 8
```

6. Configure the maximum age parameter.

```
device(config-rpvst)# max-age 22
```

7. Specify the transmit hold count.

```
device(config-rpvst)# transmit-holdcount 5
```

   This command configures the maximum number of BPDUs transmitted per second. The range of values is 1 through 10.

8. Configure VLANs.
   a. Configure VLAN 100 with a priority of 0.

```
device(config-rpvst)# vlan 100 priority 0
```

      Valid priority values range from 0 through 61440 in multiples of 4096.
   b. Configure VLAN 201 with a priority of 12288.

```
device(config-rpvst)# vlan 201 priority 12288
```

   c. Configure VLAN 301 with a priority of 20480.

```
device(config-rpvst)# vlan 301 priority 20480
```

9. Set the switching characteristics for interface 0/3.
   a. Enter interface configuration mode.

```
device(config-rpvst)# interface ethernet 0/3
```

   b. Set the switching characteristics of the interface.

```
device(conf-if-eth-0/3)# switchport
```

   c. Set the interface mode to trunk.

```
device(conf-if-eth-0/3)# switchport mode trunk
```

   d. Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
```

   e. Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
```

   f. Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
```

   g. Enable spanning tree on the interface.

```
device(conf-if-eth-0/3)# no spanning-tree shutdown
```

   h. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# exit
```

10. Set the switching characteristics for interface 0/4.
   a. Enter interface configuration mode.

```
device(config-rpvst)# interface ethernet 0/4
```

   b. Set the switching characteristics of the interface.

```
device(conf-if-eth-0/4)# switchport
```

   c. Set the interface mode to trunk.

```
device(conf-if-eth-0/4)# switchport mode trunk
```

   d. Add VLAN 100 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
```

   e. Add VLAN 201 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
```

   f. Add VLAN 301 as a member VLAN.

```
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
```

g. Enable spanning tree on the interface.

```
device(conf-if-eth-0/4)# no spanning-tree shutdown
```

h. Return to privileged EXEC mode.

```
device(conf-if-eth-0/4)# exit
```

11. To interoperate with switches other than VDX switches in R-PVST+ mode, you must configure the interface that is connected to that switch.
    a. Enter interface configuration mode for the port that interoperates with a VDX switch.

```
device(config)# interface ethernet 0/12
```

    b. Specify the MAC address for the device.

```
device(conf-if-eth-0/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
```

    c. Enable spanning tree on the interface.

```
device(conf-if-eth-0/12)# no spanning-tree shutdown
```

    d. Return to privileged EXEC mode.

```
device(conf-if-eth-0/12)# end
```

12. Verify the configuration.

```
device# show spanning-tree

VLAN 1

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 0001.01e0.5200.0180 (self)
 Bridge Id: 0001.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
```

```
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 100

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 0064.01e0.5200.0180 (self)
 Bridge Id: 0064.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 201

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 30c9.01e0.5200.0180 (self)
 Bridge Id: 30c9.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
```

```
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0

VLAN 301

 Spanning-tree Mode: Rapid PVST Protocol

 Root Id: 512d.01e0.5200.0180 (self)
 Bridge Id: 512d.01e0.5200.0180

 Root Bridge Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 20; Hello Time: 8; Max Age: 22; Max-hops: 20
 Tx-HoldCount 5
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec

 Port Et 0/3 enabled
    Ifindex: 201351168; Id: 8001; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0
```

```
 Port Et 0/4 enabled
    Ifindex: 201359360; Id: 8002; Role: Disabled; State: Disabled
    Designated Path Cost: 0
    Configured Path Cost: 20000000
    Designated Port Id: 0; Port Priority: 128
    Designated Bridge: 0000.0000.0000.0000
    Number of forward-transitions: 0
    Version: Per-VLAN Spanning Tree Protocol - Received None - Sent STP
    Portfast: off
    Configured Root guard: off; Operational Root guard: off
    Bpdu-guard: off
    Link-type: point-to-point
    Received BPDUs: 0; Sent BPDUs: 0
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

13. Save the configuration.

```
device# copy running-config startup-config
```

Enable R-PVST+ on a system configuration example

```
device# configure terminal
device(config)# protocol spanning-tree rpvst
device(config-rpvst)# bridge-priority 4096
device(config-rpvst)# forward-delay  20
device(config-rpvst)# hello-time 8
device(config-rpvst)# max-age 22
device(config-rpvst)# transmit-holdcount 5
device(config-rpvst)# vlan 100 priority 0
device(config-rpvst)# vlan 201 priority 12288
device(config-rpvst)# vlan 301 priority 20480
device(config-rpvst)# interface ethernet 0/3
device(conf-if-eth-0/3)# switchport
device(conf-if-eth-0/3)# switchport mode trunk
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/3)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/3)# no spanning-tree shutdown
device(conf-if-eth-0/3)# exit
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# switchport
device(conf-if-eth-0/4)# switchport mode trunk
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 100
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 201
device(conf-if-eth-0/4)# switchport trunk allowed vlan add 301
device(conf-if-eth-0/4)# no spanning-tree shutdown
device(conf-if-eth-0/4)# end
device# show spanning-tree
device# copy running-config startup-config
```

Clearing spanning tree counters

1. Clear spanning tree counters on all interfaces.

```
device# clear spanning-tree counter
```

2. Clear spanning tree counters on a specified Ethernet interface.

```
device# clear spanning-tree counter interface ethernet 0/3
```

3. Clear spanning tree counters on a specified port channel interface.

```
device# clear spanning-tree counter interface port-channel 12
```

Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

```
device# clear spanning-tree detected-protocols
```

2. Restart the spanning tree migration process on a specific Ethernet interface.

```
device# clear spanning-tree detected-protocols interface ethernet 0/3
```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

Port channel interface numbers range from 1 through 64.

Shutting down PVST+ or R-PVST+

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down PVST+ or R-PVST+.
   - Shut down PVST+ or R-PVST+ globally and return to privileged EXEC mode.

```
device(config)# protocol spanning-tree pvst
device(config-pvst)# shutdown
device(config-pvst)# end
```

   - Shut down PVST+ or R-PVST+ on a specific interface and return to privileged EXEC mode.

```
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# spanning-tree shutdown
device(conf-if-eth-0/2)# end
```

- Shut down PVST+ or R-PVST+ on a specific VLAN. and return to privileged EXEC mode.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
```

3. Verify the configuration.

```
device# show spanning-tree
device#
```

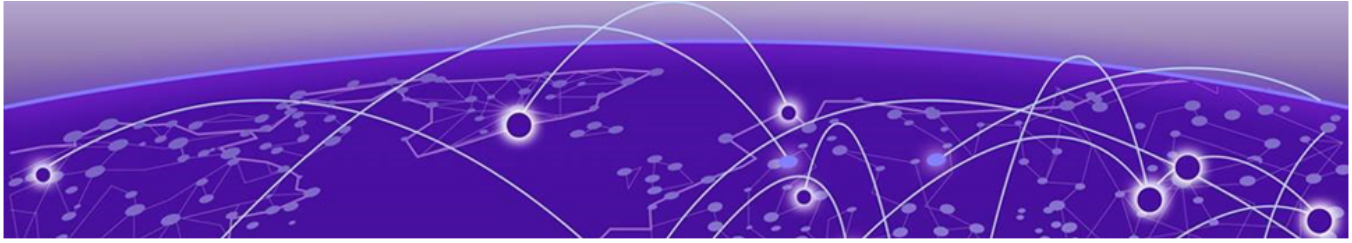
4. Save the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

Shut down PVST+ or R-PVST+ configuration example

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-vlan-10)# end
device# show spanning-tree
device# copy running-config startup-config
```

Note
Shutting down PVST+ on a VLAN is used in this example.



802.1s Multiple Spanning Tree Protocol

MSTP overview on page 244
MSTP global level parameters on page 247
MSTP interface level parameters on page 247
Configuring MSTP on page 249

MSTP overview

MSTP uses RSTP to group VLANs into separate spanning-tree instance. Each instance has its own spanning-tree topology independent of other spanning tree instances, which allows multiple forwarding paths, permits load balancing, and facilitates the movement of data traffic. A failure in one instance does not affect other instances. By enabling the MSTP, you are able to more effectively utilize the physical resources present in the network and achieve better load balancing of VLAN traffic.

The MSTP evolved as a compromise between the two extremes of the RSTP and R-PVST+, it was standardized as IEEE 802.1s and later incorporated into the IEEE 802.1Q-2003 standard. The MSTP configures a meshed topology into a loop-free, tree-like topology. When the link on a bridge port goes up, an MSTP calculation occurs on that port. The result of the calculation is the transition of the port into either a forwarding or blocking state. The result depends on the position of the port in the network and the MSTP parameters. All the data frames are forwarded over the spanning tree topology calculated by the protocol.

Note

Multiple switches must be configured consistently with the same MSTP configuration to participate in multiple spanning tree instances. A group of interconnected switches that have the same MSTP configuration is called an MSTP region.

MSTP is backward compatible with the STP and the RSTP.

Common Spanning Tree (CST)

The single Spanning Tree instance used by the Extreme device, and other vendor devices to interoperate with MSTP bridges. This spanning tree instance stretches across the entire network domain (including PVST, PVST+ and MSTP regions). It is associated with VLAN 1 on the Extreme device.

Internal Spanning Tree (IST)

An MSTP bridge must handle at least these two instances: one IST and one or more MSTIs (Multiple Spanning Tree Instances). Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance known as IST, which extends CST inside the MST region. IST always exists if the device runs MSTP. Besides IST, this implementation supports up to 31 MSTIs.

Common Internal Spanning Tree (CIST)

The single spanning tree calculated by STP (including PVST+) and RSTP (including R-PVST+) and the logical continuation of that connectivity through MSTP bridges and regions, calculated by MSTP to ensure that all LANs in the bridged LAN are simply and fully connected

Multiple Spanning Tree Instance (MSTI)

One of a number of spanning trees calculated by the MSTP within an MST Region, to provide a simply and fully connected active topology for frames classified as belonging to a VLAN that is mapped to the MSTI by the MST configuration table used by the MST bridges of that MST region.

The Extreme implementation supports up to 32 spanning tree instances in an MSTP enabled bridge that can support up to 32 different Layer 2 topologies. The spanning tree algorithm used by the MSTP is the RSTP, which provides quick convergence.

By default all configured VLANs including the default VLAN are assigned to and derive port states from CIST until explicitly assigned to MSTIs.

MST regions

MST regions are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels. Also, one or more VLANs can be mapped to one MST instance (IST or MSTI) but a VLAN cannot be mapped to multiple MSTP instances

MSTP regions

MSTP introduces a hierarchical way of managing device domains using regions. Devices that share common MSTP configuration attributes belong to a region. The MSTP configuration determines the MSTP region where each device resides. The common MSTP configuration attributes are as follows:

- Alphanumeric configuration name (32 bytes)
- Configuration revision number (2 bytes)
- 4096-element table that maps each of the VLANs to an MSTP instance

Region boundaries are determined by the above attributes. An MSTI is an RSTP instance that operates inside an MSTP region and determines the active topology for the set of VLANs mapping to that instance. Every region has a CIST that forms a single spanning tree instance which includes all the devices in the region. The difference between the CIST instance and the MSTP instance is that the CIST instance operates across the MSTP region and forms a loop-free topology across regions, while the MSTP instance operates only within a region. The CIST instance can operate using the RSTP only if all the devices across the regions support the RSTP. However, if any of the devices operate using the STP, the CIST instance reverts to the STP.

Each region is viewed logically as a single STP or a single RSTP bridge to other regions.

Note
Extreme supports 32 MSTP instances and one MSTP region.

For more information about spanning trees, see the introductory sections in the Spanning Tree Protocol chapter.

MSTP guidelines and restrictions

Follow these restrictions and guidelines when configuring the MSTP:

- Create VLANs before mapping them to the MSTP instances.
- The Extreme implementation of the MSTP supports up to 32 MSTP instances and one MSTP region.
- The MSTP **`force-version`** option is not supported.
- You must create VLANs before mapping them to the MSTP instances.
- For two or more switches to be in the same the MSTP region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same region name.
- MSTP is backward compatible with the STP and the RSTP.
- Only one MSTP region can be configured on a bridge.
- A maximum of 4090 VLANs can be configured across the 32 MSTP instances.
- MSTP and topology groups cannot be configured together.
- MSTP configured over MCT VLANs is not supported.

Default MSTP configuration

As well as the defaults listed in the section Understanding the default STP configuration on page 190, there are defaults that apply only to MSTP configurations.

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

Interoperability with PVST+ and R-PVST+

Since Extreme or other vendor devices enabled with PVST+ and R-PVST+ send IEEE STP BPDUs in addition to the PVST and R-PVST BPDUs, the VLAN 1 spanning tree joins the Common Spanning Tree (CST) of the network and thus interoperates with MSTP. The IEEE compliant devices treat the BPDUs addressed to the Extreme proprietary multicast MAC address as an unknown multicast address and flood them over the active topology for the particular VLAN.

MSTP global level parameters

To configure a switch for MSTP, first you set the region name and the revision on each switch that is being configured for MSTP. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all switches that interoperate with the same VLAN assignments.

Each of the steps used to configure and operate MSTP are described in the following:

Note
The MSTP Region and Revision global parameters are enabled for interface level parameters as described below.

- Set the MSTP region name — Each switch that is running MSTP is configured with a name. It applies to the switch which can have many different VLANs that can belong to many different MSTP regions. The default MSTP name is "NULL".
- Set the MSTP revision number — Each switch that is running MSTP is configured with a revision number. It applies to the switch, which can have many different VLANs that can belong to many different MSTP regions.
- Enabling and disabling Cisco interoperability — While in MSTP mode, use the **`cisco-interoperability`** command to enable or disable the ability to interoperate with certain legacy Cisco switches. If Cisco interoperability is required on any switch in the network, then all switches in the network must be compatible, and therefore enabled by means of this command. By default the Cisco interoperability is disabled.
- The parameters you would normally set when you configure STP are applicable to MSTP. Before you configure MSTP parameters see the sections explaining bridge parameters, the error disable timeout parameter and the port-channel path cost parameter in the STP section of this guide.

MSTP interface level parameters

Edge port and automatic edge detection

From an interface, you can configure a device to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

Follow these guidelines to configure a port as an edge port:

- When edge port is enabled, the port still participates in a spanning tree.
- A port can become an edge port if no BPDU is received.
- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

Note
If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a **shutdown** or **no shutdown** command.

BPDU guard

In a valid configuration, edge port-configured interfaces do not receive BPDUs. If an edge port-configured interface receives a BPDU, an invalid configuration exists, such as the connection of an unauthorized device. The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service.

BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in an STP topology change. In this case, you can enable the STP BPDU guard feature on the Extreme device port to which the end station is connected. The STP BPDU guard shuts down the port and puts it into an "error disabled" state. This disables the connected device's ability to initiate or participate in an STP topology. A log message is then generated for a BPDU guard violation, and a message is displayed to warn the network administrator of an invalid configuration.

The BPDU Guard provides a secure response to invalid configurations because the administrator must manually put the interface back in service with the **no shutdown** command if error disable recovery is not enabled by enabling the **errdisable-timeout** command. The interface can also be automatically configured to be enabled after a timeout. However, if the offending BPDUs are still being received, the port is disabled again.

Expected behavior in an interface context

When BPDU Guard is enabled on an interface, the device is expected to put the interface in Error Disabled state when BPDU is received on the port when edge-port and BPDU guard is enabled on the switch interface. When the port ceases to receive the BPDUs, it does not automatically switch to edge port mode, you must configure **error disable timeout** or **no shutdown** on the port to move the port back into edge port mode.

Restricted role

Restricted role ports are selected as an alternate port after the root port has been selected. It is configured by a network administrator to prevent bridges external to a core region of the network influencing the spanning tree active topology, possibly because those bridges are not under the full control of the administrator. It will protect the root bridge from malicious attack or even unintentional misconfigurations where a bridge device which is not intended to be root bridge, becomes root bridge causing severe bottlenecks in data path. These types of mistakes or attacks can be avoided by configuring 'restricted-role' feature on ports of the root bridge . This feature is similar to the "root-guard" feature which is proprietary implementation of Cisco for STP and RSTP but had been adapted in the 802.1Q standard as "restricted-role". The "restricted-role" feature if configured on an incorrect port can cause lack of spanning tree connectivity.

Expected behavior in an interface context

When this feature is enabled on an interface the device is expected to prevent a port configured with restricted-role feature from assuming the role of a Root port. Such a port is expected to assume the role of an Alternate port instead, once Root port is selected.

Restricted TCN

Configuring "restricted TCN" on a port causes the port not to propagate received topology change notifications and topology changes originated from a bridge external to the core network to other ports. It is configured by a network administrator to prevent bridges external to a core region of the network from causing MAC address flushing in that region, possibly because those bridges are not under the full control of the administrator for the attached LANs. If configured on an incorrect port it can cause temporary loss of connectivity after changes in a spanning trees active topology as a result of persistent incorrectly learned station location information.

Expected behavior in an interface context

When this feature is enabled on an interface, the device is expected to prevent propagation of topology change notifications from a port configured with the Restricted TCN feature to other ports. In this manner, the device prevents TCN propagation from causing MAC flushes in the entire core network.

Configuring MSTP

Enabling and configuring MSTP globally

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable MSTP.

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

   This command creates a context for MSTP. MSTP is automatically enabled. All MSTP specific CLI commands can be issued only from this context. Entering **no protocol spanning-tree mstp** deletes the context and all the configurations defined within the context.
3. Specify the region name.

```
device(config-mstp)# region kerry
```

4. Specify the revision number.

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

5. Configure an optional description of the MSTP instance.

```
device(config-mstp)# description kerry switches
```

6. Specify the maximum hops for a BPDU to prevent the messages from looping indefinitely on the interface.

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

   Setting this parameter prevents messages from looping indefinitely on the interface. The range is 1 through 40 hops while the default is 20 .
7. Map VLANs to MSTP instances and set the instance priority.
   a. Map VLANs 7 and 8 to instance 1.

```
device(config-mstp)# instance 1 vlan 7,8
```

   b. Map VLANs 21, 22, and 23 to instance 2.

```
device(config-mstp)# instance 2 vlan 21-23
```

   c. Set the priority of instance 1.

```
device(config-mstp)# instance 1 priority 4096
```

   This command can be used only after the VLAN is created. VLAN instance mapping is removed from the configuration if the underlying VLANs are deleted.
8. Configure a bridge priority for the CIST bridge.

```
device(config-mstp)# bridge-priority 4096
```

   The range is 0 through 61440 in increments of 4096. The default is 32768.
9. Set the error disable parameters.
   a. Enable the timer to bring the port out of error disable state.

```
device(config-mstp)# error-disable-timeout enable
```

b. Specify the time in seconds it takes for an interface to time out.

```
device(config-mstp)# error-disable-timeout interval 60
```

The range is from 10 to 1000000 seconds with a default of 300 seconds.

10. Configure forward delay.
    a. Specify the bridge forward delay.

```
device(config-mstp)# forward-delay 15
```

This command allows you to specify how long an interface remains in the listening and learning states before it begins forwarding. This command affects all MSTP instances. The range of values is from 4 to 30 seconds with a default of 15 seconds.

11. Configure hello time.

```
device(config-mstp)# hello-time 2
```

The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The range is from 1 through 10 seconds with a default of 2 seconds.

12. Configure the maximum age.

```
device(config-mstp)# max-age 20
```

You must set the **max-age** so that it is greater than the **hello-time**. The range is 6 through 40 seconds with a default of 20 seconds.

13. Specify the port-channel path cost.

```
device(config-mstp)# port-channel path-cost custom
```

This command allows you to control the path cost of a port channel according to bandwidth.

14. Specify the transmit hold count.

```
device(config-mstp)# transmit-holdcount 5
```

The transmit hold count is used to limit the maximum number of MSTP BPDUs that the bridge can transmit on a port before pausing for 1 second. The range is from 1 to 10 seconds with a default of 6 seconds.

15. Configure Cisco interoperability.

```
device(config-mstp)# cisco-interoperability enable
```

This command enables the ability to interoperate with certain legacy Cisco switches. The default is Cisco interoperability is disabled.

16. Return to privileged exec mode.

```
device(config-mstp)# end
```

17. Verify the configuration. The following is an example configuration.

```
device# show spanning-tree mst-config

Spanning-tree Mode: Multiple Spanning Tree Protocol
```

```
CIST Root Id: 8000.001b.ed9f.1700
CIST Bridge Id: 8000.768e.f80a.6800
CIST Reg Root Id: 8000.001b.ed9f.1700

CIST Root Path Cost: 0; CIST Root Port: Eth 1/2
CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 19
Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20;
Tx-HoldCount: 6
Number of topology change(s): 139; Last change occurred 00:03:36 ago on Eth 1/2

Bpdu-guard errdisable timeout: disabled
Bpdu-guard errdisable timeout interval: 300 sec
Migrate Time: 3 sec

 Name           : kerry
 Revision Level : 1
 Digest         : 0x9357EBB7A8D74DD5FEF4F2BAB50531AA

 Instance        VLAN
 --------        ----
  0:             1
  1:             7,8
  2:             21-23
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

18. Save the configuration.

```
device# copy running-config startup-config
```

MSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# region kerry
device(config-mstp)# revision 1
device(config-mstp)# description kerry switches
device(config-mstp)# max-hops 20
device(config-mstp)# instance 1 vlan 7,8
device(config-mstp)# instance 2 vlan 21-23
device(config-mstp)# instance 1 priority 4096
device(config-mstp)# bridge-priority 4096
device(config-mstp)# error-disable-timeout enable
device(config-mstp)# error-disable-timeout interval 60
device(config-mstp)# forward-delay 16
device(config-mstp)# hello-time 5
device(config-mstp)# max-age 16
device(config-mstp)# port-channel path-cost custon
device(config-mstp)# transmit-holdcount 5
device(config-mstp)# cisco-interoperability enable
device(config-mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Enabling and configuring MSTP on an interface

The parameters can be configured individually on an interface by:

1. Entering the commands in Steps 1 through Step 3 for the target interface
2. Running the relevant parameter command
3. Verifying the result
4. Saving the configuration

For detailed descriptions of the parameters and features, see the sections STP parameters and STP features.

1. Enter configuration mode.

```
device# configure terminal
```

2. Enable MSTP.

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

3. Enter interface configuration mode.

```
device(config-mstp)# interface ethernet 0/5
```

4. Enable the interface.

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

5. Configure the restricted role feature for the port.

```
device(conf-if-eth-0/5)# spanning-tree restricted-role
```

This command keeps a port from becoming a root.

6. Restrict topology change notifications (TCN) BPDUs for an MSTP instance.

```
device(conf-if-eth-0/5)# spanning-tree instance 5 restricted-tcn
```

This prevents the port from propagating received TCNs and topology changes originating from a bridge, external to the core network, to other ports.

7. Enable auto detection of an MSTP edge port.

```
device(conf-if-eth-0/5)# spanning-tree autoedge
```

Enabling this feature allows the system to automatically identify the edge port. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

8.

```
device(conf-if-eth-0/5)# spanning-tree edgeport
```

Enabling edge port allows the port to quickly transition to the forwarding state. By default, automatic edge detection is disabled.

9. Enable BPDU guard on the port

```
device(conf-if-eth-0/5)# spanning-tree edgeport bpdu-guard
```

   BPDU guard removes a node that reflects BPDUs back in the network. It enforces the STP domain borders and keeps the active topology predictable by not allowing any network devices behind a BPDU guard-enabled port to participate in STP.

10. Set the path cost of a port.

```
device(conf-if-eth-0/5)# spanning-tree cost 200000
```

    The path cost range is from 1 to 200000000. Leaving the default adjusts path cost relative to changes in the bandwidth. A lower path cost indicates greater likelihood of becoming root port.

11. Configure the link type.

```
device(conf-if-eth-0/5)# spanning-tree link-type point-to-point
```

    The options are **point-to-point** or **shared**.

12. Enable port priority.

```
device(conf-if-eth-0/5)# spanning-tree priority 128
```

    The range is from 0 to 240 in increments of 16 with a default of 32. A lower priority indicates greater likelihood of becoming root port.

13. Return to privileged exec mode.

```
device(conf-if-eth-0/5)# end
```

14. Verify the configuration.

```
device# show spanning-tree interface ethernet 0/5

Spanning-tree Mode: Multiple Spanning Tree Protocol

 Root Id: 8000.001b.ed9f.1700
 Bridge Id: 8000.01e0.5200.011d

 Port Eth 0/5 enabled
    Ifindex: 411271175; Id: 8002; Role: Designated; State: Forwarding
    Designated External Path Cost: 0; Internal Path Cost: 20000000
    Configured Path Cost: 200000
    Designated Port Id: 8002; Port Priority: 128
    Designated Bridge: 8000.01e0.5200.011d
    Number of forward-transitions: 1
    Version: Multiple Spanning Tree Protocol - Received MSTP - Sent MSTP
    Edgeport: yes; AutoEdge: yes; AdminEdge: no; EdgeDelay: 3 sec
    Restricted-role is enabled
    Restricted-tcn is enabled
    Boundary: no
    Bpdu-guard: on
    Link-type: point-to-point
    Received BPDUs: 86; Sent BPDUs: 1654
```

15. Save the configuration.

```
device# copy running-config startup-config
```

Enable MSTP on an interface configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# interface ethernet 0/5
device(conf-if-eth-0/5)# no shutdown
device(conf-if-eth-0/5)# spanning-tree restricted-role
device(conf-if-eth-0/5)# spanning-tree instance 5 restricted-tcn
device(conf-if-eth-0/5)# spanning-tree autoedge
device(conf-if-eth-0/5)# spanning-tree edgeport
device(conf-if-eth-0/5)# spanning-tree edgeport bpdu-guard
device(conf-if-eth-0/5)# spanning-tree cost 200000
device(conf-if-eth-0/5)# spanning-tree link-type point-to-point
device(conf-if-eth-0/5)# spanning-tree priority 128
device(conf-if-eth-0/5)# end
device# show spanning-tree interface ethernet 0/5
device# copy running-config startup-config
```

Enabling MSTP on a VLAN

1. Enter configuration mode.

```
device# configure terminal
```

2. Enter the protocol command to enable MSTP configuration.

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

3. Map a VLAN to an MSTP instance.

```
device(config-mstp)# instance 5 vlan 300
```

4. Return to privileged EXEC mode.

```
device(config-mstp)# end
```

5. Verify the configuration.

```
device# show spanning-tree mst

 Spanning-tree Mode: Multiple Spanning Tree Protocol

 CIST Root Id: 8000.609c.9f5d.4800 (self)
 CIST Bridge Id: 8000.609c.9f5d.4800
 CIST Reg Root Id: 8000.609c.9f5d.4800 (self)

 CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20;
 Tx-HoldCount: 6
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec
 Migrate Time: 3 sec

  Name           : NULL
  Revision Level : 0
  Digest         : 0xD5FF4C3F6C18E2F27AF3A8300297ABAA

  Instance        VLAN
  --------        ----
```

```
    0:              1
    5:              100
```

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

6. Save the configuration.

```
device# copy running-config startup-config
```

Enable spanning tree on a VLAN configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# instance 5 vlan 300
device(config-mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Configuring basic MSTP parameters

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable MSTP.

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

3. Specify the region name.

```
device(config-mstp)# region connemara
```

4. Specify the revision number.

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

5. Map MSTP instances to VLANs.
   a. Map instance 1 to VLANs 2 and 3.

```
device(config-mstp)# instance 1 vlan 2,3
```

   b. Map instance 2 to VLANs 4, 5, and 6.

```
device(config-mstp)# instance 2 vlan 4-6
```

6. Set a priority for an instance.

```
device(conf-Mstp)# instance 1 priority 28672
```

The priority ranges from 0 through 61440 and the value must be in multiples of 4096.

7. Specify the maximum hops for a BPDU.

```
device(conf-Mstp)# max-hops 25
```

   This prevents the messages from looping indefinitely on an interface
8. Return to privileged EXEC mode.

```
device(conf-Mstp)# end
```

9. Verify the configuration.

```
device# show spanning-tree mst

 Spanning-tree Mode: Multiple Spanning Tree Protocol

 CIST Root Id: 8000.609c.9f5d.4800 (self)
 CIST Bridge Id: 8000.609c.9f5d.4800
 CIST Reg Root Id: 8000.609c.9f5d.4800 (self)

 CIST Root Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 20
 Configured Forward Delay: 15; Hello Time: 2; Max Age: 20; Max-hops: 25;
 Tx-HoldCount: 6
 Number of topology change(s): 0

 Bpdu-guard errdisable timeout: disabled
 Bpdu-guard errdisable timeout interval: 300 sec
 Migrate Time: 3 sec

  Name           : connemara
  Revision Level : 1
  Digest         : 0xD5FF4C3F6C18E2F27AF3A8300297ABAA

  Instance        VLAN
  --------        ----
   0:             1,7,8,9
   1:             2,3
   2:             4-6
```

Note

Observe that the settings comply with the formula set out in the STP parameters section, as:

(2 × (forward delay - 1)) ≥ maximum age ≥ (2 × (hello time + 1))

or in this case: 28 ≥ 20 ≥ 6.

```
device# show running-config | begin spanning-tree
protocol spanning-tree mstp
 instance 1 vlan 2,3
 instance 1 priority 28672
 instance 2 vlan 4-6
 region connemars
 revision 1
 max-hops 25
 !
...
```

10. Save the configuration

```
device# copy running-config startup-config
```

Basic MSTP configuration example

```
device# configure terminal
device(config)# protocol spanning-tree mstp
device(config-mstp)# region connemara
device(config-mstp)# revision 1
device(config-mstp)# instance 1 vlan 2,3
device(config-mstp)# instance 2 vlan 4-6
device(conf-Mstp)# instance 1 priority 28582
device(conf-Mstp)# max-hops 25
device(conf-Mstp)# end
device# show spanning-tree mst
device# copy running-config startup-config
```

Clearing spanning tree counters

1. Clear spanning tree counters on all interfaces.

```
device# clear spanning-tree counter
```

2. Clear spanning tree counters on a specified Ethernet interface.

```
device# clear spanning-tree counter interface ethernet 0/3
```

3. Clear spanning tree counters on a specified port channel interface.

```
device# clear spanning-tree counter interface port-channel 12
```

Port channel interface numbers range from 1 through 64.

Clearing spanning tree-detected protocols

These commands force a spanning tree renegotiation with neighboring devices on either all interfaces or on a specified interface.

1. Restart the spanning tree migration process on all interfaces.

```
device# clear spanning-tree detected-protocols
```

2. Restart the spanning tree migration process on a specific Ethernet interface.

```
device# clear spanning-tree detected-protocols interface ethernet 0/3
```

3. Restart the spanning tree migration process on a specific port channel interface.

```
device# clear spanning-tree detected-protocols port-channel 12
```

Port channel interface numbers range from 1 through 64.

Shutting down MSTP

1. Enter global configuration mode.

```
device# configure terminal
```

2. Shut down MSTP.
   - Shut down MSTP globally and return to privileged EXEC mode.

     ```
     device(config)# protocol spanning-tree mstp
     device(config-mstp)# shutdown
     device(config-mstp)# end
     ```

   - Shut down MSTP on a specific interface and return to privileged EXEC mode.

     ```
     device(config)# interface ethernet 0/2
     device(conf-if-eth-0/2)# spanning-tree shutdown
     device(conf-if-eth-0/2)# end
     ```

   - Shut down MSTP on a specific VLAN and return to privileged EXEC mode.

     ```
     device(config)# vlan 10
     device(config-vlan-10)# spanning-tree shutdown
     device(config-vlan-10)# end
     ```

3. Verify the configuration.

   ```
   device# show spanning-tree
   device#
   ```

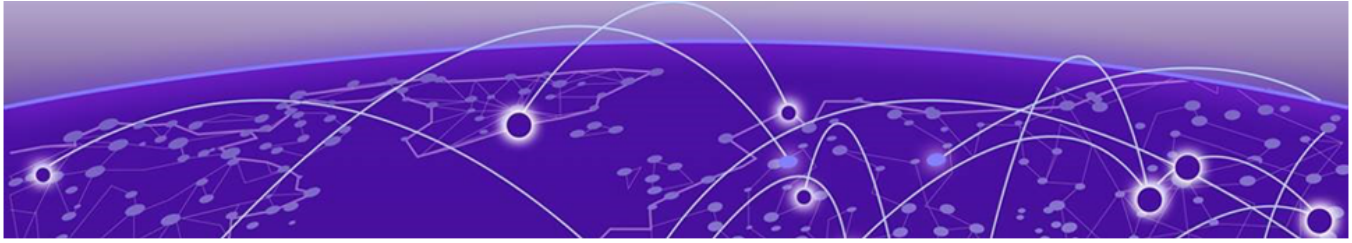
4. Save the running configuration to the startup configuration.

   ```
   device# copy running-config startup-config
   ```

Shut down MSTP configuration example

```
device# configure terminal
device(config)# vlan 10
device(config-vlan-10)# spanning-tree shutdown
device(config-stp)# end
device# show spanning-tree
device# copy running-config startup-config
```

Note
Shutting down MSTP on a VLAN is used in this example.



Topology Groups

Topology groups on page 260
Master VLAN, member VLANs, and bridge-domains on page 260
Control ports and free ports on page 261
Configuration considerations on page 261
Configuring a topology group on page 262
Displaying topology group information on page 264

Topology groups

A topology group is a named set of VLANs and bridge-domains that share a Layer 2 control protocol. Topology groups simplify configuration and enhance scalability of Layer 2 protocols by allowing you to run a single instance of a Layer 2 protocol on multiple VLANs and bridge-domains. One instance of the Layer 2 protocol controls all the VLANs and bridge-domains.

You can use topology groups with the following Layer 2 protocols:

- Per VLAN Spanning Tree (PVST+)
- Rapid per VLAN Spanning tree (R-PVST+)

Master VLAN, member VLANs, and bridge-domains

Each topology group contains a master VLAN and can contain one or more member VLANs and bridge-domains. A definition for each of these VLAN types follows:

- Master VLAN—The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for Rapid per VLAN Spanning tree (R-PVST), the topology group's master VLAN contains the R-PVST configuration information.
- Member VLANs—The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the member VLANs. Member VLANs do not independently run a Layer 2 protocol.
- Member bridge domains—The member bridge domains are similar to VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the bridge domains. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the bridge domains. Bridge domains do not independently run a Layer 2 protocol. In a

bridge domain, a single port can have multiple logical interfaces. In this scenario, all the logical interfaces on that port (and bridge domain) will follow the state of master VLAN port.

When a Layer 2 topology change occurs, resulting in a change of port state in the master VLAN, the same port state is applied to all the member VLANs and bridge-domains belonging to the topology group on that port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs and bridge-domains that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Control ports and free ports

A port in a topology group can be a control port or a free port:

- A **control port** is a port in the master VLAN and, therefore, is controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs and bridge-domains is controlled by the master VLAN's Layer 2 protocol. Each member VLAN and bridge-domain must contain all of the control ports. All other ports in the member VLAN and bridge-domain are "free ports."
- **Free ports** are not controlled by the master VLAN's Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs and bridge-domains that are not also in the master VLAN are free ports.

Note
Because free ports are not controlled by the master port's Layer 2 protocol, they are always in the forwarding state.

Configuration considerations

The configuration considerations are as follows:

- You can configure up to 128 topology groups. A VLAN or bridge-domain cannot controlled by more than one topology group. You can configure up to 4K VLANs or bridge-domain as members of topology group.
- The topology group must contain a master VLAN. The group can also contain individual member VLANs and or member bridge-domains. You must configure the member VLANs or member bridge-domains before adding them to the topology group. Bridge-domains cannot be configured as a master VLAN.
- You cannot delete a master VLAN from the topology group when the member VLANs or bridge-domains are in the topology group.
- The control port membership must match the master VLAN when adding a member VLAN or member bridge-domain.
- If a VLAN enabled with the PVST+ or R-PVST+ protocol is added as a member VLAN of a topology group, the protocol is disabled. The member VLAN is added to the topology group. If the VLAN is removed from the topology group, the protocol is disabled, and you must enable the protocol if required.

- Enabling STP on an interface is only allowed if both master VLAN and member VLAN or bridge-domains are configured on the interface across all topology groups.
- You cannot remove the master VLAN or member VLAN or bridge-domains from an STP enabled interface.
- Topology group configuration is allowed only with PVST+ and R-PVST+ spanning tree configurations.

Configuring a topology group

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

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

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

```
device(config)# topology-group 1
device(conf-topo-group-1)#
```

Note
The **no topology-group** command deletes an existing topology group.

Configuring a master VLAN

Before configuring a master VLAN, you should have configured a topology group.

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

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

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

```
device(config)# topology-group 1
device(conf-topo-group-1)#
```

3. Enter the **master-vlan** command to configure a master VLAN in the topology group.

```
device(conf-topo-group-1)# master-vlan 100
```

Note
The **no master-vlan** command removes an existing master VLAN from the topology group.

Adding member VLANs

Before adding a member VLAN, you should have created a topology group and configured the master VLAN for that group. The VLAN should not be part of any

other topology group. All control ports of master VLAN must also be configured for the member VLAN.

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

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

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

```
device(config)# topology-group 1
device(conf-topo-group-1)#
```

3. Enter the **master-vlan** command to configure a master VLAN in the topology group.

```
device(conf-topo-group-1)# master-vlan 100
```

4. Enter the **member-vlan** command to add member VLANs to the topology group.

```
device(conf-topo-group-1)# member-vlan add 200-201
```

Note

The **member-vlan remove** command removes an existing member VLAN from the topology group.

```
device(conf-topo-group-1)# member-vlan remove 200
```

Adding member bridge-domains

Before adding a bridge domain, you should have created a topology group and configured the master VLAN for that group. The bridge-domain should not be part of any other topology group. All control ports of master VLAN must also be configured for the member bridge-domain.

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

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

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

```
device(config)# topology-group 1
device(conf-topo-group-1)#
```

3. Enter the **master-vlan** command to configure a master VLAN in the topology group.

```
device(conf-topo-group-1)# master-vlan 100
```

4. Enter the **member-bridge-domain** command to add member bridge-domains to the topology group.

```
device(conf-topo-group-1)# member-bridge-domain add 300
```

Note
The **member-bridge-domain remove** command removes an existing member bridge-domain from the topology group.

```
device(conf-topo-group-1)# member-bridge-domain remove 1
```

The example adds 300 as member bridge-domain to the topology group.

```
device# configure terminal
device(config)# topology-group 1
device(conf-topo-group-1)# master-vlan 100
device(conf-topo-group-1)# member-bridge-domain add 300
```

Replacing a master VLAN

To avoid temporary loops when the master VLAN is replaced by another VLAN, the following recommendation is made:

- Control ports for both the old and the new master VLAN must match.
- The new master VLAN and the old master VLAN must have same ports in the blocking state to avoid the possibility of temporary loops.

If the recommendation is not followed, and a new master VLAN is configured with a different convergence, the configuration is still accepted.

Note
The master VLAN replacement is accepted if both the old and the new master VLANs are spanning-tree disabled.

Displaying topology group information

Before displaying the topology group information, you should have configured a topology group and defined the master VLAN.

Enter the **show topology-group** command to display the group information.

```
device# show topology-group 1
Topology Group 1
==============================
Master VLAN : 100
L2 Protocol: R-PVST
Member VLANs : 200 300
Member Bridge-domains: 10
Control Ports : eth 2/1, eth 2/2, po10
Free Ports : VLAN: 200 -eth 2/3, po11
Bridge-domain: 10 -eth 2/3.20, po11.10
```

The example displays information about topology group 1.

The **show running-config** command displays topology group configurations.

```
device# show running-config
topology-group 1
   master-vlan 100
   member-vlan add 200 300
   member-bridge-domain add 10
```

Loop Detection

LD protocol overview on page 266
LD use cases on page 272
Configuring LD protocol on page 275
Loop detection for VLAN on page 277

LD protocol overview

Layer 2 networks rely on learning and flooding to build their forwarding databases. Because of the flooding nature of these networks, any loops can be disastrous as they cause broadcast storms.

Important
The LD feature should be used only as a tool to detect loops in the network. It should not be used to replace other Layer 2 protocols such as STP.

This feature provides support for the following:

- Strict and loose modes
- Multi-Chassis Trunk (MCT)
- Breakout ports
- EPVN VLAN tunnels

LD protocol data units (PDUs) are initiated and received on the native device. Loop detection and action on the port state is also done on the same native device. Intermediate devices in the network must be capable of flooding unknown Layer 2 unicast PDUs on the VLAN through which they are received.

Strict mode

In what is referred to as *strict mode*, LD is configured on an interface. If the LD PDU is sent on an interface and received on the same interface, that port is shut down by LD. Strict mode overcomes specific hardware issues that cause packets to be echoed back to the input port. The following figure illustrates strict mode.

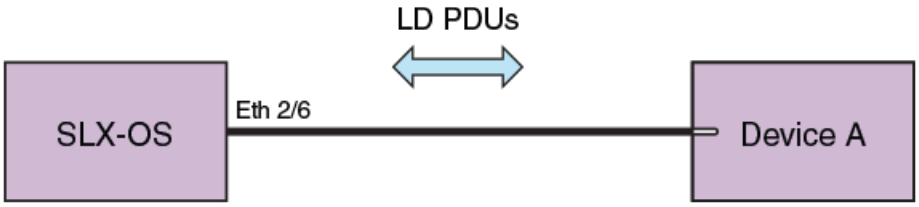


Figure 28: Strict mode

If the user provides a VLAN, then the PDUs are tagged accordingly. Otherwise PDUs are sent untagged. With a LAG, PDUs are sent out on the port-channel interface. If Device A has a loop (for example, a LAG is not configured), then the PDU is flooded back to SLX-OS, which detects the loop. In case of a loop, the port-channel interface is shut down. The following figure illustrates LD on a LAG.

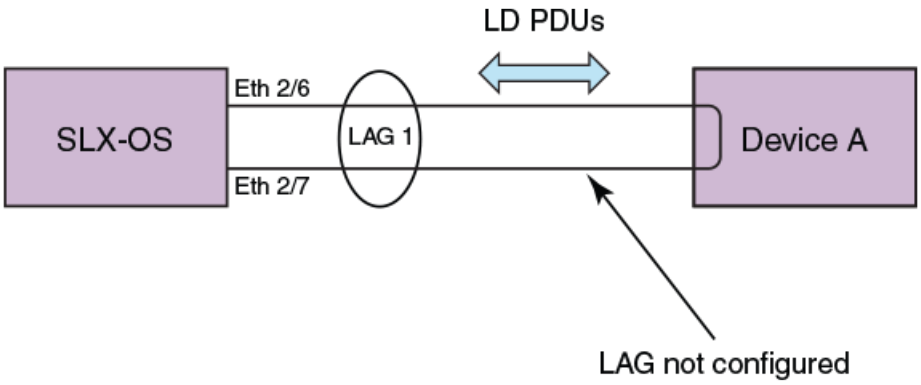


Figure 29: LD on a LAG

LD supports 256 instances of strict mode.

Loose mode

In what is referred to as *loose mode*, LD is configured on a VLAN. If a VLAN in the device receives an LD PDU that originated from the same device on that VLAN, this is considered to be a loop and the receiving port is shut down. In loose mode, LD works at the VLAN level and takes action at the logical interface (LIF) level. The following figure illustrates loose mode, with LD on a VLAN.

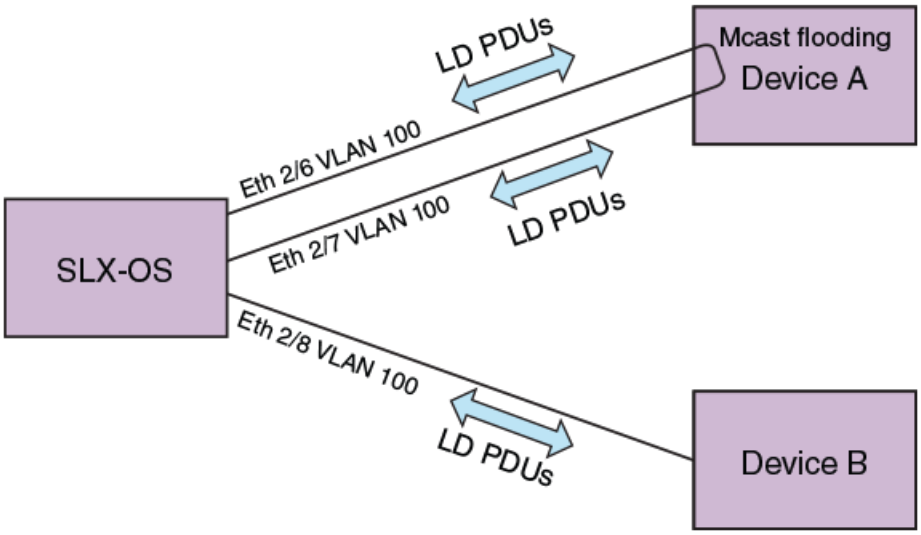


Figure 30: Loose mode: LD on a VLAN

SLX-OS generates the LD PDUs on the VLAN. if Device A has a loop, PDUs are flooded back to SLX-OS, which detects the loop. SLX-OS then shuts down the receiving LIF of the port on the VLAN.

LD supports 256 instances of loose mode, which means that it can be enabled on 256 VLANs.

LD PDU format

The following figure illustrates the format of the LD PDU in bytes.

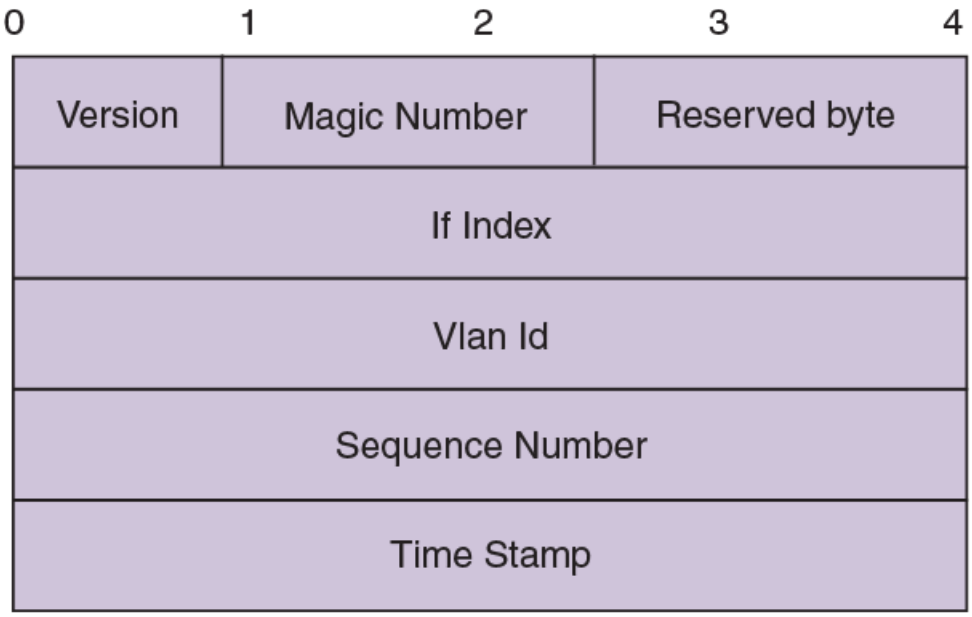


Figure 31: LD PDU format in bytes

Parameter	Definition
Version	LD protocol version (1 by default)
Magic Number	0x13EF; used to differentiate between LD multicast PDUs and other multicast PDUs
Reserved byte	For future use
If Index	Index of the source port; populated only in strict mode
Vlan Id	VLAN ID
Sequence Number	Reserved for future enhancements
Time Stamp	Reserved for future enhancements

LD PDU transmission

Each LD-enabled interface or VLAN on a device continually transmits Layer 2 LD PDUs at a 1-second default hello-timer interval, with the destination MAC address as the multicast address. The multicast MAC address is derived from the system MAC address of the device with the multicast bit (8) and the local bit (7) set.

For example, if the MAC address is 00E0.5200.1800, then the multicast MAC address is 03E0.5200.1800. In the case of a LAG port-channel, LD PDUs are sent out one of the ports of the LAG as chosen by hardware.

LD PDU reception

When the LD PDU is received and is generated by the same device, the PDU is processed. If the PDU is generated by another device, then the PDU is flooded.

If a port is already blocked by any other Layer 2 protocol such as STP, then the LD PDUs are neither sent for LD processing nor flooded on that port.

LD parameters

This section discusses the various global protocol-level, interface level, and VLAN-level parameters that are used to control and process LD PDUs.

Protocol level

hello-interval

hello-interval is the rate at which the LD PDUs are transmitted by an LD-enabled interface or VLAN, which is 1000 milliseconds by default. Lowering the hello-interval below the default increases the PDU transmission rate, providing faster loop detection and also removing transient loops that last less than one second. On the other hand, increasing the interval above the default (for example, to 100 milliseconds) can increase the steady-state CPU load.

shutdown-time

shutdown-time is the duration after which an interface that is shut down by LD is automatically reenabled. The range is from 0 through 1440 minutes. The default is 0 minutes, which means that the interface is not automatically reenabled.

Important
Changing this value can cause repeated interface flapping when a loop is persistent in the network.

raslog-duration

raslog-duration is the interval between RASLog messages when a port is shut down by LD to prevent flooding of these messages. The range is from 10 through 1440 minutes. The default is 10.

Interface level

In strict mode, the parameters in this section are configurable at the interface level, and the configuration is specific to an interface. The following figure illustrates strict mode configuration.

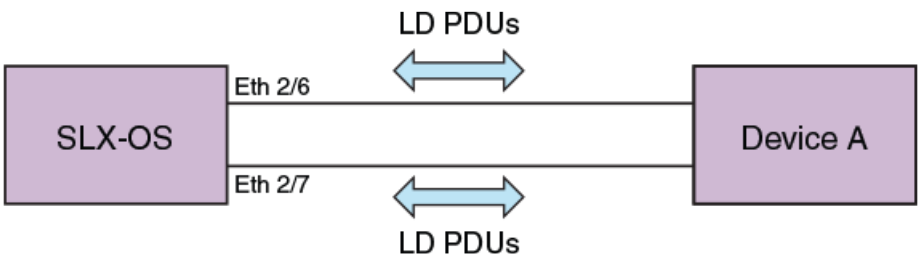


Figure 32: Strict mode configuration

shutdown-disable

By default, the device shuts down the interface if a loop is detected. Configuring **shutdown-disable** means that the interface shutdown is disabled and LD never brings down such interface. If a loop is already detected by LD and the port is in shutdown state, then configuring **shutdown-disable** is not effective until the port is back up.

vlan-association

Although user can enable LD on an interface without specifying a VLAN, the **vlan-association** keyword is used to specify a VLAN associated with the interface.

VLAN level

In loose mode, the user can configure LD under a VLAN. In this case, LD PDUs are flooded on the VLAN. The following figure illustrates loose mode configuration.

Figure 33: Loose mode configuration

LD PDU processing

As long as LD PDUs are not received, there is no loop. If an LD PDU is received, then there is a loop that is present in the network.

If the if-index field in the received LD PDU is valid, then it is considered to be operating in strict mode. If the port on which the LD PDU was received is same as one encoded in the PDU (with a match for VLAN ID if a VLAN is associated), the port is shut down. For an MCT, if a strict mode LD PDU is received on an ICL interface, and the PDU is originated by another interface, then the ICL interface is not shut down. Instead, the sender interface is shut down. In addition, for strict mode the required interfaces should be configured with LD, or else the PDUs will not get processed

If the if-index field in the received LD PDU is invalid, then it is considered to be operating in loose mode. Based on VLAN ID information present in the received LD PDU, the receiving LIF is shut down. If the receiving interface is an MCT ICL interface, the LD PDU is dropped.

In the case of an LD-enabled LAG (port-channel) interface, if the sent LD PDU is received on the port-channel, then the port-channel interface is shut down.

If the **shutdown-disable** option is configured for the particular interface, then the port drops the received PDU without processing it.

The re-enablement of the LD shut down port depends on the **shutdown-time** configuration. For manual recovery, either flap the interface, by means of the **shutdown**

and **no shutdown** commands, or clear the loop by means of the **clear loop-detection** command.

Support for EPVN VLAN tunnels

LD loose mode is used to support a shutdown at the attachment circuit (AC) logical interface (LIF) level instead of at the physical port level. See Loop detection for VLAN on page 277 and configuration examples.

Configuration considerations

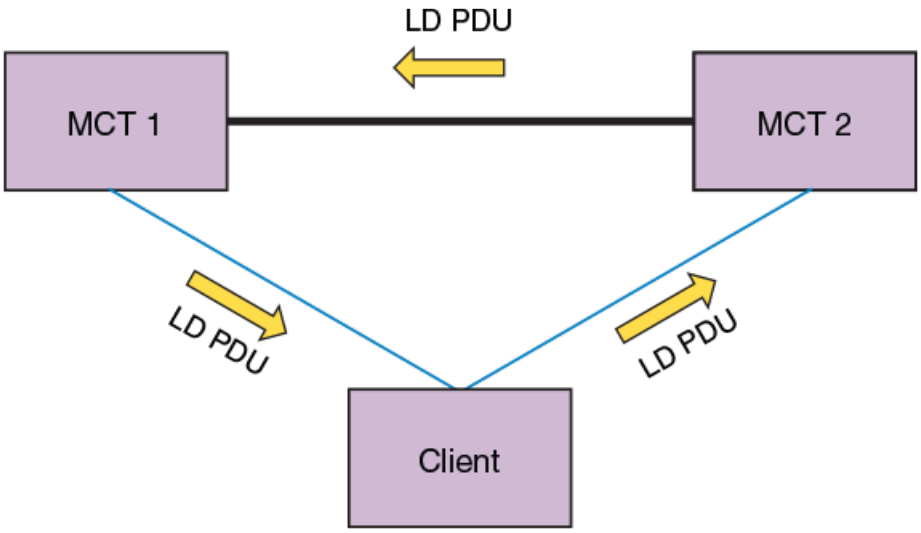
On an external switch that is unaware of LD or where LD is not configured, there may be some ACL rules applied to interfaces to permit traffic from known MAC addresses, and at the last of these rules there is an ACL deny-any rule to block all unknown MAC addresses. If this interface is part of a loop, LD enabled on SLX-OS will not be able to detect and break the loop.

LD use cases

In an MCT configuration, LD runs independently on both nodes. With loose mode the user must enable loop detection for the same VLANs on both nodes in the MCT cluster. MCT strict mode and loose mode use cases are detailed below.

MCT strict mode

Strict mode LD is enabled on the MCT 1 cluster client edge port (CCEP) interface that connects to the Client.



1. MCT 1 generates LD PDUs.
2. If the Client has the LAG interface configured to support LD, the Client drops the PDUs and there is no loop.
3. If there is a misconfiguration, the Client floods the PDUs, reaching MCT 2.
4. MCT 1 then identifies the interface information encoded in the PDUs, shutting down the interface on which the packets were generated.

Figure 34: MCT strict mode

MCT loose mode

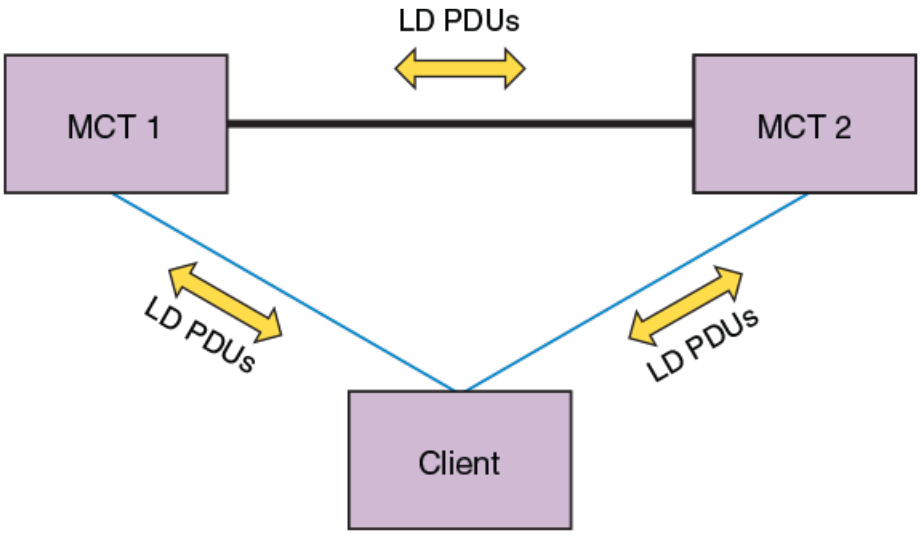


Figure 35: MCT loose mode: Use case 1

Use case 1: LD enabled on VLAN *x* on MCT 1

1. MCT 1 sends LD PDUs on VLAN x on all the interfaces that are part of the CCEP, client edge port (CEP), and ICL interface.
2. If the Client has LD configured on the LAG interface, then it drops the PDUs and no loop exists. If there is a misconfiguration, the Client floods the PDUs and they reach MCT 2.
3. MCT 2 floods the PDUs back to MCT 1, where the loop is detected. With loose mode no information about the interface that transmitted the PDU is encoded in the PDU, so normally the receiving interface is shut down. Because in this case the PDU is received on the ICL interface, that interface is not shut down.
4. MCT 1 receives the loop detection PDUs on the CCEP interface as well, as the packets were flooded in the VLAN in the following sequence: MCT 1 > MCT 2 > Client > MCT 1. In this case the receiving CCEP is shut down to break the loop. For MCT 2 to forward the PDUs in this case it must be the designated forwarder (DF) for that VLAN.

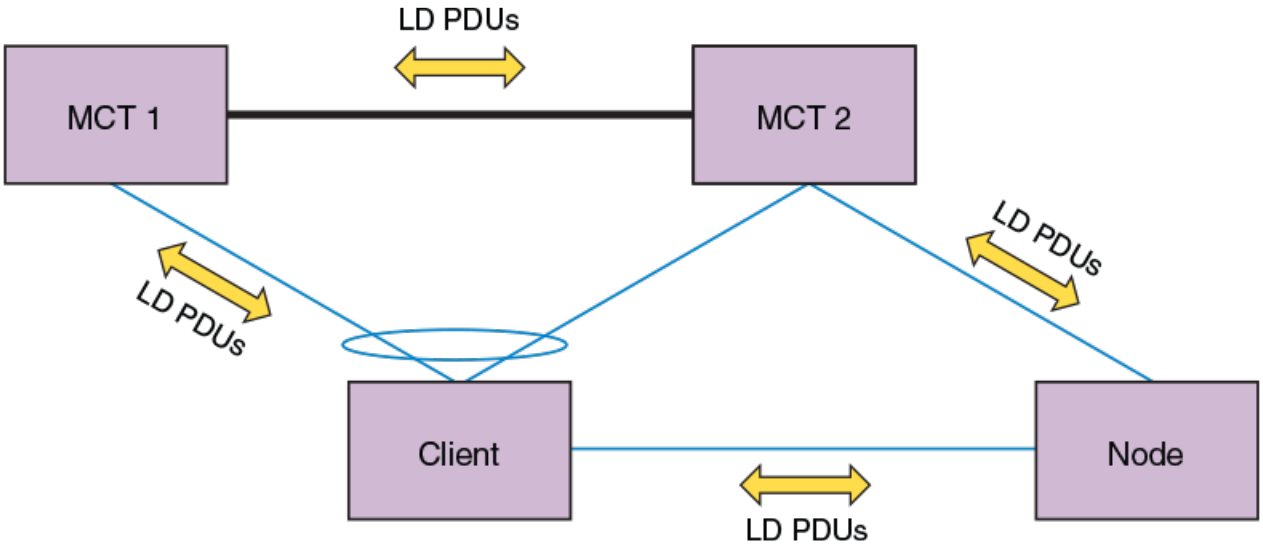


Figure 36: MCT loose mode: Use case 2

Use case 2: LD enabled on VLAN *x* on MCT 1 and MCT 2

1. Both MCT 1 and MCT 2 will flood the PDUs in VLAN *x* on all the interfaces that are part of the CCEP, CEP, and ICL interface.
2. Assuming PDUs from MCT 1 take the path MCT 1 > MCT 2 > Node > Client > MCT 1, then the receiving CCEP interface is shut down. For MCT 2 to forward the PDUs in this case, it must be the DF for that VLAN.
3. Assuming PDUs from MCT 2 take the path MCT 2 > MCT 1 > Client > Node > MCT 2, then the receiving CEP interface is shut down
4. If PDUs from MCT 2 take the path MCT 2 > Node > Client > MCT 2, then the receiving CCEP interface is shut down.
5. Multiple interfaces can be shut down in this case, depending on the sequence in which loops are detected.
6. In addition, to avoid CCEP interfaces from being shut down over a CEP interface, the user can configure a CCEP port not to be shut down.

Configuring LD protocol

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **protocol loop-detection** command to enable loop detection, enter Protocol Loop Detect configuration mode, and configure a variety of global options.

```
device(config)# protocol loop-detection
```

3. (Optional) Enter the **hello-interval** command to change the hello interval from the default.

```
device(config-loop-detect)# hello-interval 2000
```

4. (Optional) Enter the **shutdown-time** command to change from the default the interval after which an interface that is shut down by loop detection (LD) protocol is automatically reenabled.

```
device(config-loop-detect)# shutdown-tiome 20
```

5. (Optional) Enter the **raslog-duration** command to change from default the interval between RASLog messages that are sent when a port is disabled by the loop detection (LD) protocol.

```
device(config-loop-detect)# raslog-duration 20
```

6. Enable LD at the interface level.
   a. In global configuration mode, specify an interface (either an Ethernet interface or a port-channel interface).

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

   b. In interface subtype configuration mode, enter the **loop-detection** command.

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

7. Enable LD at the VLAN level.
   a. In global configuration mode, create a VLAN.

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

   b. In VLAN configuration mode, enter the **loop-detection** command.

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

8. Associate the VLAN with an interface.
   a. In global configuration mode, specify an interface (either an Ethernet interface or a port-channel interface).

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

   b. In interface subtype configuration mode, enter the **loop-detection vlan** command and specify a VLAN. (The VLAN must already be created.)

```
device(conf-if-eth-0/6)# loop-detection vlan 5
```

9. (Optional) Disable the shutting down of an interface (Ethernet or port-channel) as a result of the loop detection (LD) protocol.
   a. In global configuration mode, specify an interface (either an Ethernet interface or a port-channel interface).

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

b. In interface subtype configuration mode, enter the **loop-detection shutdown-disable** command.

```
device(conf-if-eth-0/6)# loop-detection shutdown-disable
```

10. (Optional) Disable the shutting down of an interface (Ethernet or port-channel) as a result of the loop detection (LD) protocol.
    a. In global configuration mode, specify an interface (either an Ethernet interface or a port-channel interface).

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

    b. In interface subtype configuration mode, enter the **loop-detection shutdown-disable** command.

```
device(conf-if-eth-0/6)# loop-detection shutdown-disable
```

11. Confirm the LD configuration, using the **show loop-detection** command with a variety of options.
    a. To display LD information at the system level, enter the **show loop-detection** command as in the following example.

```
device# show loop-detection
Strict Mode:
------------------------

Number of loop-detection instances enabled: 1

Interface: eth 0/6
        Enabled on VLANs: 100
        Shutdown Disable:   No
        Interface status: UP
        Auto enable in: Never

Packet Statistics:
vlan            sent            rcvd        disable-count
100             100             0           0

Loose Mode:
------------------------

Number of LD instances:   2
Disabled Ports:                   2/7

Packet Statistics:
vlan            sent            rcvd        disable-count
100             100             0           0
```

    b. To display ports disabled by LD, enter the **show loop-detection disabled-ports** command as in the following example.

```
device# show loop-detection disabled-ports
Ports disabled by loop detection
-----------------------------------------
port       age(min)       disable cause
0/6        5              Disabled by Self
```

    c. To display global LD configuration values, enter the **show loop-detection globals** command.

```
device# show loop-detection globals
Loop Detection:                Disabled
```

```
Shutdown-time (minutes):        0
Hello-time (msec):              1000
Raslog-duration (minutes):      10
```

12. Use the **clear loop-detection** command in privileged EXEC mode with a variety of options to reenable ports that were disabled by LD and clear the LD statistics.
    a. To enable LD-disabled ports and clear LD statistics on all interfaces, enter the **clear loop-detection** command.

    ```
    device# clear loop-detection
    ```

    b. To enable LD-disabled ports and clear LD statistics on an Ethernet interface, enter the **clear loop-detection interface ethernet** command.

    ```
    device# clear loop-detection interface ethernet 0/6
    ```

    c. To enable LD-disabled ports and clear LD statistics on a port-channel interface, enter the **clear loop-detection interface port-channel** command.

    ```
    device# clear loop-detection interface port-channel 20
    ```

    d. To enable LD-disabled ports and clear LD statistics on a VLAN, enter the **clear loop-detection interface vlan** command.

    ```
    device# clear loop-detection interface vlan 10
    ```

Loop detection for VLAN

Note
Loop detection is not supported for bridge domains (BDs).

When a loop is detected on a VLAN and port, only the LIF of the VLAN on the port is shut down, but the physical port still remains up and other VLANs on the port are not affected.

The existing LD loose mode configuration commands support loop detection for VLAN tunnels. In LD loose mode, if the VLAN is mapped to VLAN tunnels and LD is enabled, VLAN tunnels loop detection is supported. Up to 256 LD loose mode instances can be configured.

If a loop is detected from a VLAN tunnel, the following actions can take place:

- A RASLog is sent and the tunnel VNI LIFs on which the loop is detected is shut down.
- A RASLog is sent but the tunnel LIF is not shut down.

Configuring loop detection for VLAN

The following example enables loop detection on a VLAN and enters Protocol Loop Detection configuration mode.

```
device# configure terminal
device(config)# vlan 5
device(config-vlan-5)# loop-detection
device(config-loop-detect)#
```

The following example enables ports associated with VLAN 8 and clears LD statistics for that VLAN.

```
device# clear loop-detection vlan 8
```

The following example displays LD configuration values, including logical interfaces (LIFs), for a VLAN tunnel.

```
device# show loop-detection vlan 20
Number of LD instances: 1
LIF (Logical Interface) Disabled on Ports: eth2/2

Packet Statistics:
vlan            sent            rcvd
20              119             1
```

The following example displays LD configuration values for a VLAN tunnel if LD shutdown is disabled.

```
device# show loop-detection vlan 20
Number of LD instances: 1
LIF (Logical Interface) ShutDown is disabled for VLAN 20

Packet Statistics:
vlan            sent            rcvd
20              10              10
```

The following example disables the shutdown of a VLAN VLAN tunnel.

```
device# configure terminal
device(config)# vlan 20
device(config-vlan-20)# loop-detection-shutdown-disable
```

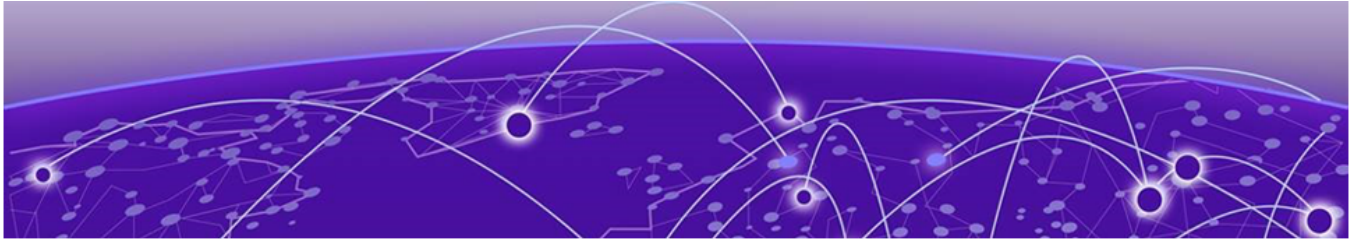
You can control the auto-enable behavior of an LD-disabled logical interface (LIF), by using the **shutdown-time** command in Protocol Loopback Detection configuration mode. The following example specifies a shutdown time of 1 minute.

```
device# configure terminal
device(config)# loop-detection
device(config-loop-detect)# shutdown-time 1
2017/10/20-16:04:48, [ELD-1005], 3749, M2 | Active | DCE, INFO, SLX, Loop is detected on Ethernet 2/2
VLAN 20, the LIF (logical interface) is shutdown.
2017/10/20-16:05:46, [ELD-1007], 3750, M2 | Active | DCE, INFO, SLX, Loop detection disabled LIF
(Logical interface) on Ethernet 2/2 VLAN 20 is auto-enabled.
```

By default the shutdown time is 0, which means that an LD-disabled LIF is never auto-enabled. If the shutdown time is configured with a nonzero value, the LD-disabled LIF is auto-enabled following the specified shutdown time.

To enable LD-disabled ports and clear LD statistics on all interfaces, use the **clear loop-detection** command as in the following example.

```
device# clear loop-detection
```



Ethernet Ring Protection Protocol

Ethernet Ring Protection overview on page 279
Initializing a new ERN on page 284
Signal Fail on page 289
Manual Switch on page 290
Forced Switch on page 293
Dual-end blocking on page 297
Non-revertive mode on page 298
Interconnected rings on page 298
Configuring ERP on page 299

Ethernet Ring Protection overview

Ethernet Ring Protection (ERP), a nonproprietary protocol described in ITU-T G.8032 (Version 1 and 2), integrates an Automatic Protection Switching (APS) protocol and protection switching mechanisms to provide Layer 2 loop avoidance and fast reconvergence in Layer 2 ring topologies. ERP supports multi-ring and ladder topologies. ERP can also function with IEEE 802.1ag to support link monitoring when non-participating devices exist within the Ethernet ring.

Note
Before configuring ERP, you must configure a VLAN and the ports you require for your deployment.

This chapter describes ERP components, features, and how to configure, and manage ERP.

Configuration Considerations

- ERP is supported on the SLX-9540 and the SLX-9640 devices.
- You can have a maximum of 255 ERP instances on a device.
- Changes to a master VLAN apply to the member VLANs.
- If a Topology Group is used, then an ERP instance can be configured only for the Master VLAN.
- VLANs used for ERP must be pre-configured on the device along with interfaces in L2 mode.

- ERP cannot be configured if STP/RSTP/MSTP/PVST is running, and STP/RSTP/MSTP/PVST cannot be configured if ERP is running.
- Sub-50 msec convergence is achieved with a 4-device Ring topology on 1-RU devices. If the number of nodes in the topology increases beyond this, there will be a small linear increase in convergence time.
- Maintenance Domain (MD), Maintenance Association (MA), and Maintenance End Points (MEPs) need to be configured before using **dot1ag** compliance for a specific ERP instance.

ERP components

An ERP deployment consists of the following components:

- Roles assigned to devices, called Ethernet Ring Nodes (ERNs)
- Interfaces
- Protocols -- ERP alone or with IEEE 802.1ag
- ERP messaging
- ERP operational states
- ERP timers

ERN roles

In an Ethernet ring topology you can assign each ERN one of three roles:

- Ring Protection Link Owner (RPL owner) -- One RPL owner must exist in each ring; its role is to prevent loops by maintaining a break in traffic flow to one configured link while no failure condition exists within the ring.
- Non-RPL node -- Multiple non-RPL nodes, can exist in a ring; but they have no special role and perform only as ring members. Ring members apply and then forward the information received in R-APS messages.
- Ring Protection Link (RPL) node -- RPL nodes block traffic to the segment that connects to the blocking port of the RPL owner. The RPL node is used in dual-end blocking and is part of the FDB optimization feature.

Each device can only have one role at any time. Non-ERN devices can also exist in topologies that use IEEE 802.1ag.

ERN interfaces

In addition to a role, each ERN has two configured interfaces:

- Left interface
- Right interface

Traffic enters one interface (ingress) and exits the device using the other interface (egress). The right and left interfaces are physically connected.

You must configure these left and right interfaces in the same pattern across all ERNs within a topology. For example you can assign the interfaces as left/right, left/right, left/right, and so on. It is not acceptable, however, to assign interfaces in random

order, such as left/right in the configuration of one ERN and then right/left in the configuration of the next ERN.

Protocols

You can configure standalone ERP or ERP with IEEE 802.1ag support.

Using standalone ERP

When using standalone ERP, all devices have a role, and all devices participate at least as ERP members.

Ring-APS (R-APS) messages are sent at initial start-up of a configuration and periodically when link or node failures or recoveries occur. Each ERN applies the information received in the R-APS messages and forwards the received RAPS messages if both ports are in the forwarding state.

The sending ERN terminates the message when it receives a message originally sent from itself.

Configurable timers prevent ERNs from receiving outdated messages and decrease failure reporting time to allow increased stability within the topology.

To properly configure and troubleshoot ERP, an understanding of the messaging, operational states, and timers is essential. For more information about the ERP protocol, see ITU-T G.8032.

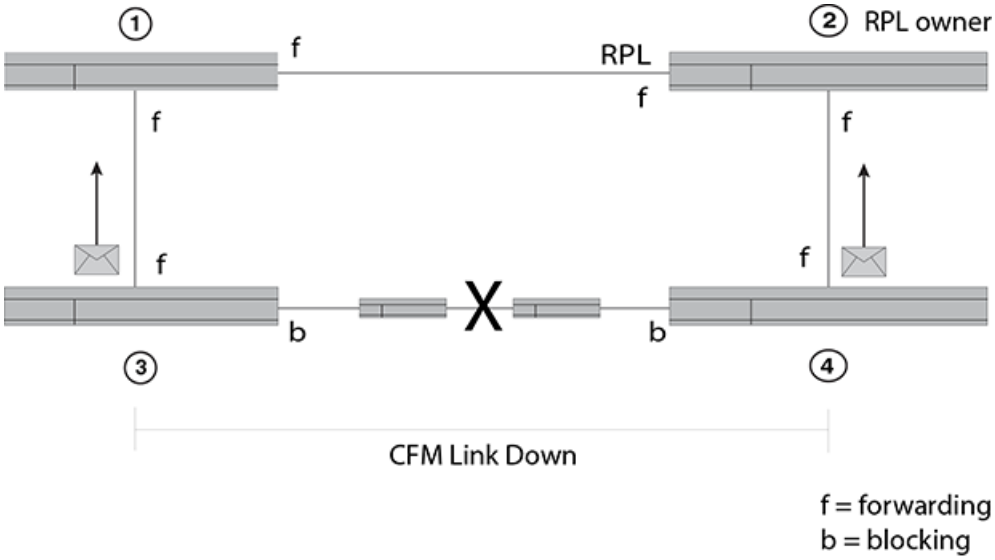
Using ERP with IEEE 802.1ag support

When you have other nonparticipating switches in the ring, you can use the IEEE 802.1ag support to perform link health checks to the next ERN.

With IEEE 802.1ag configured, the ERNs within the ring send Continuity Check Messages (CCM) to verify the integrity of their own links. If a node is not receiving CCMs or if a link goes down, a failure is reported to the ring through R-APS messages.

The following figure shows a segment with ERNs 3 and 4 and two non-participating switches located on the same network segment between them. When ERNs 3 and 4 stopped receiving CCMs, the following actions occurred on ERNs 3 and 4:

Figure 37: ERP with IEEE 802.1ag support



1. Blocked the failed port.
2. Transmitted a R-APS (SF) message.
3. Unblocked the non-failed port.
4. Flushed the FDB.
5. Entered the Protection state.

   As a result, ERN 2, the RPL owner, unblocked the RPL, and the topology became stable and loop free.

ERP messaging

In ERP, ERNs send R-APS messages. For details about the packet structures, see ITU-T G.8032.

The destination MAC address (Dst Mac) is the first element in the packet and is of the form 01-19-A7-00-00-<ERP ID>. The default value is 01. However, you can configure the ERP ID with the **raps-default-mac** command. In ITU-T G.8032 Version 1 the default value is always used.

The Node ID indicates the base MAC address and can be found in the R-APS specific information part of a R-APS message.

ERP operational states

RPL nodes can be in one of six different states in Version 2:

- Init
- Idle
- Protection state, which is designated as a Signal Fail (SF) event in the R-APS
- Manual Switch (MS)
- Forced Switch (FS)
- Pending (not available in Version 1)

When an ERP topology starts up, each ERN (in Init state) transmits a R-APS (NR). After start-up, the behavior varies by assigned role. The following table shows the initialization process for an ERN.

Message exchange and actions during ERN initialization version 2

Table 44: ERP operational states

RPL owner	Non-RPL node	RPL node
Init state	Init state	Init state
1. Blocks the RPL. 2. Sends a R-APS (NR). 3. Enters the Pending state.	1. Blocks the left interface. 2. Sends a R-APS (NR). 3. Enters the Pending state.	1. Blocks the left interface. 2. Sends a R-APS (NR). 3. Enters the Pending state.
4. Starts the WTR timer. 5. (After the WTR expires) stops sending NR. 6. Sends R-APS (NR, RB, DNF). 7. Enters the Idle state.	After receiving the (NR, RB, DNF) from the RPL owner: 1. Unblocks the non-failed blocking port. 2. Stops sending (NR). 3. Enters the Idle state.	After receiving the (NR, RB, DNF) from the RPL owner: 1. Blocks the RPL port. 2. Unblocks the other ports. 3. Enters the Idle state.

When the ring is in the Pending state, an ERN flushes the filtering database (FDB) if it receives any of the following state requests:

- Signal Fail (SF)
- No request (NR), RPL Blocked (RB)

Note
ITU-T G.8032 Version 1 does not use a Pending state, so from the Protection state ERNs enter the Idle state.

ERP timers

ERP provides various timers to ensure stability in the ring while a recovery is in progress or to prevent frequent triggering of the protection switching. All of the timers are operator configurable.

- **Guard timer** -- All ERNs use a guard timer. The guard timer prevents the possibility of forming a closed loop and prevents ERNs from applying outdated R-APS messages. The guard timer activates when an ERN receives information about a local switching request, such as after a switch fail (SF), manual switch (MS), or forced switch (FS). When this timer expires, the ERN begins to apply actions from the R-APS it receives. This timer cannot be manually stopped.
- **Wait-To-Restore (WTR) timer** -- The RPL owner uses the WTR timer. The WTR timer applies to the revertive mode to prevent frequent triggering of the protection switching due to port flapping or intermittent signal failure defects. When this timer expires, the RPL owner sends a R-APS (NR, RB) through the ring.
- **Wait-To-Block (WTB) timers** -- The WTB timer is activated on the RPL owner. The RPL owner uses WTB timers before initiating an RPL block and then reverting to

the idle state after operator-initiated commands, such as for FS or MS conditions, are entered. Because multiple FS commands are allowed to co-exist in a ring, the WTB timer ensures that the clearing of a single FS command does not trigger the re-blocking of the RPL. The WTB timer is defined to be 5 seconds longer than the guard timer, which is enough time to allow a reporting ERN to transmit two R-APS messages and allow the ring to identify the latent condition. When a MS command is cleared, the WTB timer prevents the formation of a closed loop caused by the RPL owner node applying an outdated remote MS request during the recovery process.

- **Hold-off timer** -- Each ERN uses a hold-off timer to delay reporting a port failure. When the timer expires, the ERN checks the port status. If the issue still exists, the failure is reported. If the issue does not exist, nothing is reported.
- **Message interval** -- This is an operator configurable feature for sending out R-APS messages continuously when events occur.

Initializing a new ERN

A newly configured Version 2 ERP topology with four ERNs initializes as described in this section. The ERNs have the following roles:

- ERN 2 is the RPL owner.
- ERNs 1, 3, and 4 are non-RPL nodes.

The following figure shows the first step of initialization beginning from ERN 4, a non-RPL node.

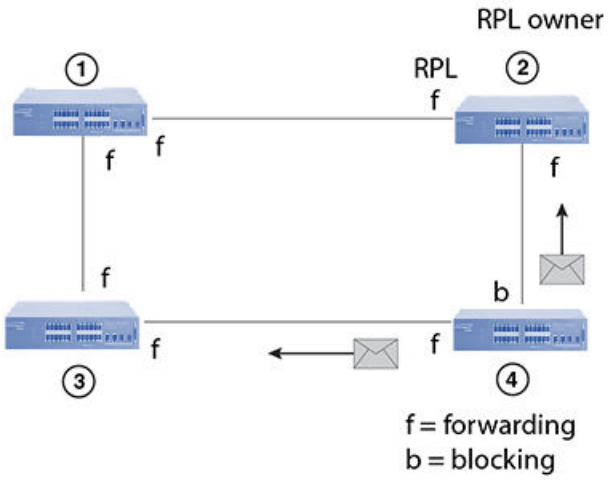


Figure 38: Initializing an ERN topology - I

The actions of each ERN are as follows:

- ERN 1 takes no action. Both ports are in the forwarding state.
- ERN 2 (RPL owner) takes no action. Both ports, including the RPL port, are in the VLAN port forwarding state.

- ERN 3 takes no action. Both ports are in the forwarding state.
- From the Init state ERN 4 stops all timers (guard, WTR, WTB), blocks the left port, unblocks the right port, transmits R-APS (NR) messages, and enters the Pending state.

The following figure shows the next sequence of events. Next, ERN 1 initializes.

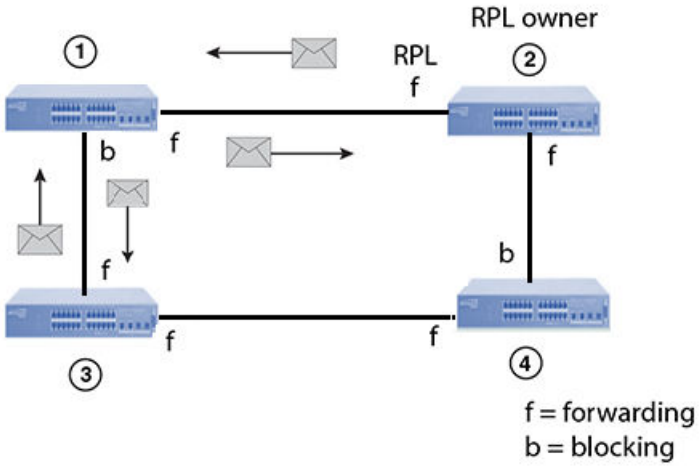


Figure 39: Initializing an ERP topology - II

The actions of each ERN are as follows:

- ERN 1 stops all timers (guard, WTR, WTB), blocks the left port, unblocks the right port, transmits R-APS (NR) messages, and enters the Pending state.
- ERN 2 takes no action. Both ports are in the forwarding state.
- ERN 3 takes no action. Both ports are in the forwarding state.
- ERN 4 stays in the Pending state, transmits R-APS (NR) messages, and continues to block the left interface.

The following figure shows the next sequence of events.

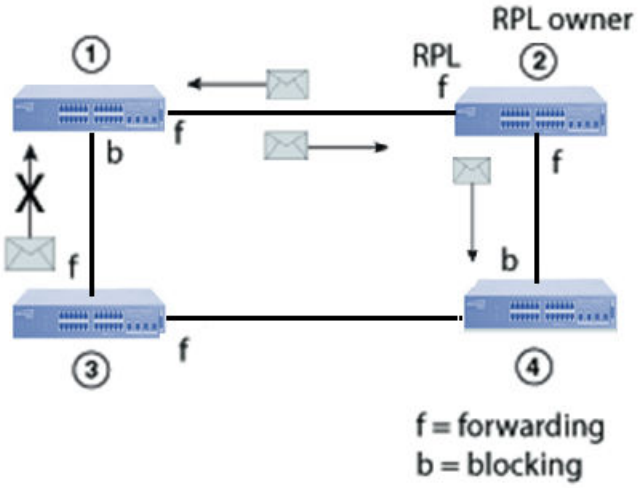


Figure 40: Initializing an ERP topology - III

The actions of each ERN are as follows:

- ERN 1 terminates R-APS received on the blocked port, unblocks the non-failed port, stops transmitting R-APS (NR) messages, and enters the Pending state.
- ERN 2 takes no action.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages.

The following figure shows the next sequence of events.

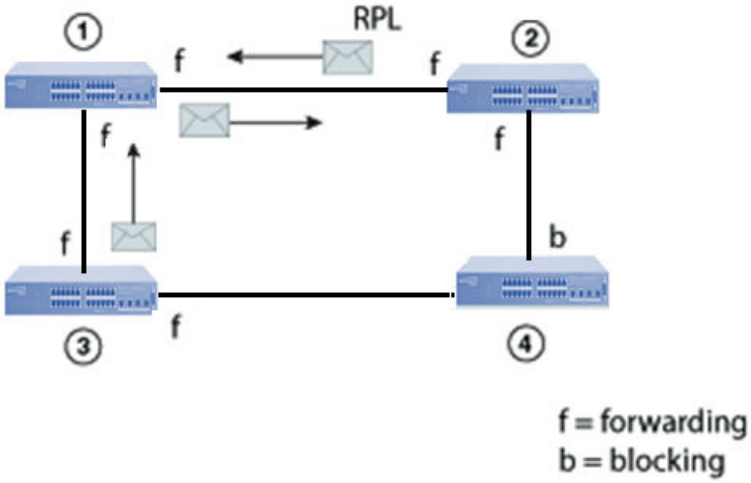


Figure 41: Initializing an ERP topology - IV

The actions of each ERN are as follows:

- ERN 1, from the Pending state, unblocks the left interface, stops sending R-APS (NR) and stays in the Pending state. Now both interfaces are in the forwarding state.
- ERN 2 takes no action.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages. The left interface is blocked, and the right interface is in the forwarding state.

The following figure shows the next sequence of events. Next ERN 2 initializes.

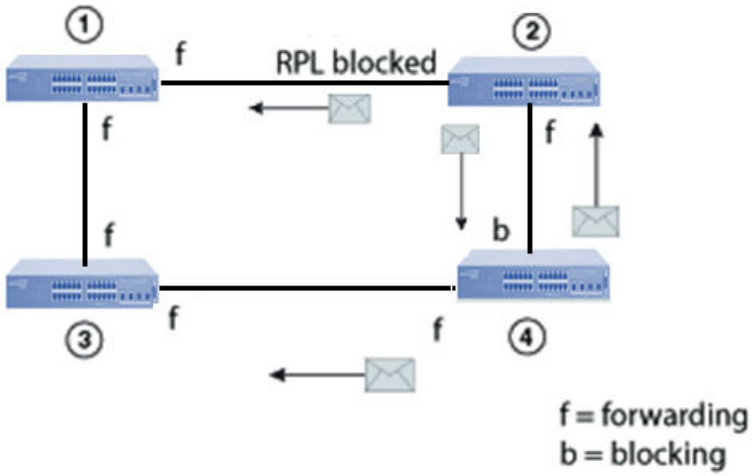


Figure 42: Initializing an ERP topology - V

The actions of each ERN are as follows:

- ERN 1 stays in the Pending state.
- ERN 2 (RPL owner), from the Init state, stops the guard timer, stops the WTB timer, blocks the RPL, unblocks the non-RPL port, enters the Pending state, transmits R-APS (NR) messages, and starts the WTR timer.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages. The left interface is blocked.

The following figure shows the next sequence of events.

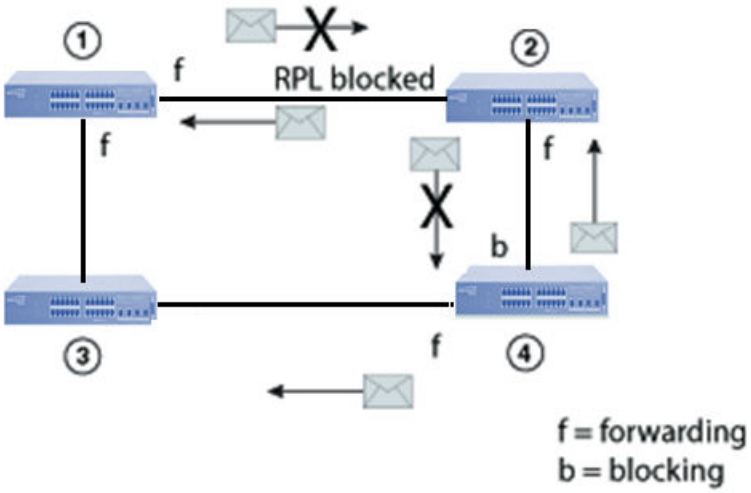


Figure 43: Initializing an ERP topology - VI

The actions of each ERN are as follows:

- After the WTB timer expires, ERN 2 (RPL owner in the Pending state) transmits R-APS (NR, RB), and then ERN 2 enters the Idle state.
- ERN 1, still in the Pending state, forwards R-APS (NR, RB) and enters the Idle state.
- ERN 3 takes no action.
- ERN 4 from the Pending state and stops transmitting R-APS (NR).

Lastly, ERNs 1, 2, and 3 are in the Idle state, and ERN 4 changes the blocking port to the forwarding state. All ERNs remain in the Idle state. Refer to the following figure.

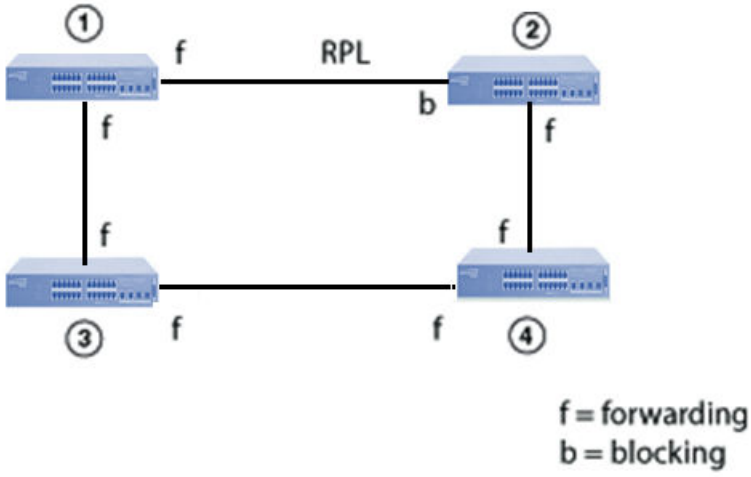


Figure 44: Initializing an ERP topology - VII

Signal Fail

Signal Fail (SF) and SF recovery provide the mechanism to repair the ring to preserve connectivity among customer networks.

ERP guarantees that although physically the topology is a ring, logically it is loop-free. One link, called the Ring Protection Link (RPL), is blocked to traffic. When a non-RPL link fails in the ring, the SF mechanism triggers and causes the RPL to become forwarding. Later, signal fail recovery can occur to restore the ring to the original setup.

Convergence time is the total time that it takes for the RPL owner to receive the R-APS (NR) message and block the RPL port until the ERN with the failed link receives notice and unblocks the failed link.

The following figure shows a simple Ethernet ring topology before a failure. This diagram shows dual-end blocking enabled (thick line) between ERNs one (RPL node) and 6 (RPL owner). ERNs 3, 2, 4, and 5 are non-RPL nodes.

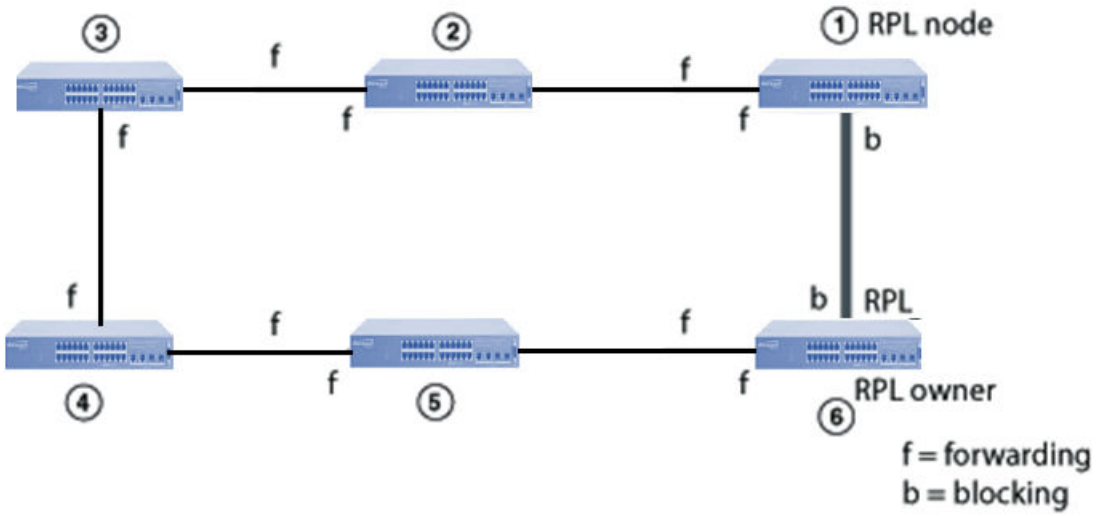


Figure 45: ERP topology

The following figure shows the same Ethernet ring topology after a failure at the forwarding port of ERN 4 when a signal fail triggered, and ring protection was needed. ERN 6 unblocked the RPL port and the RPL node changed the blocking port to the forwarding state.

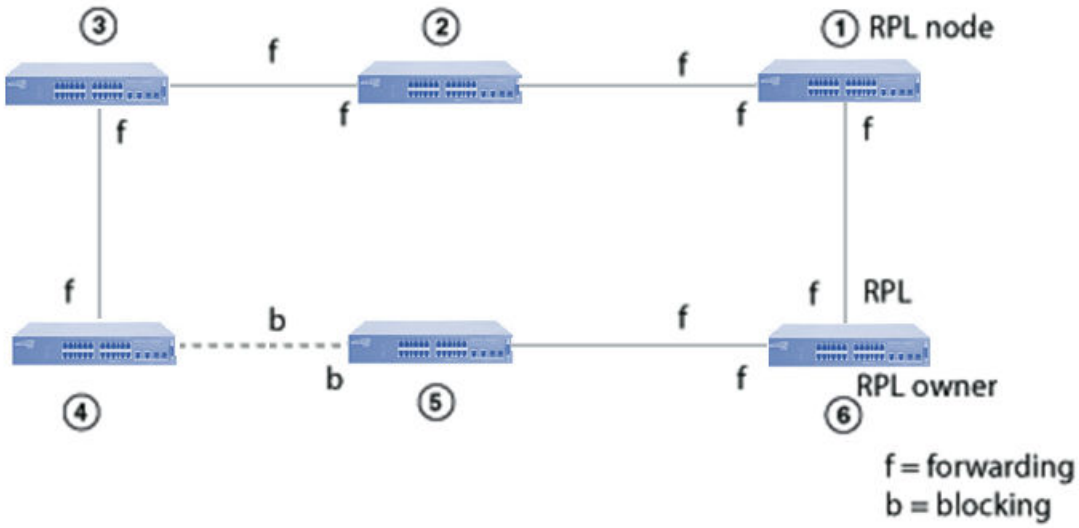


Figure 46: ERP topology in a Protected state

Manual Switch

In the absence of a failure, an operator-initiated Manual Switch (MS) moves the blocking role of the RPL by blocking a different ring link and initiates the node sending a R-APS (MS) to inform the RPL owner to unblock the RPL. This can occur if no higher priority request exists in the ring.

Consider a ring consisting of nodes ERN1, ERN2, ERN3, and ERN4. Dual-end blocking is enabled between ERN1 and ERN2.

A node that receives the R-APS (MS) forwards it to the adjacent nodes. If the receiving node is already in the Idle or Pending state, it unblocks the non-failed port and stops transmitting R-APS messages. Only one MS can exist in the topology at any time. An MS condition has to be manually cleared.

Note
If any ERN is in an FS state or in a protected state through an SF event and an operator tries to configure an MS, the ERN will reject the request.

When a manual switch is cleared by an operator on the same node on which the MS is configured, the node keeps the port in a blocking state, sends out a R-APS (NR) to the adjacent node, and starts the guard timer. Other nodes that receive the R-APS (NR) forward the message. When the RPL owner receives this message, then the RPL owner starts the WTR timer. When the WTR timer expires, the RPL owner sends out a R-APS (NR, RB), blocks the RPL, and flushes the FDB. Other nodes in the topology that receive the R-APS (NR, RB) unblock any non-failed port and flush the FDB.

In order to clear the MS condition, the operator must enter the manual switch command from ERN3. Refer to the following table for the event sequence.

Table 45: MS on Non-RPL node event sequence

Non-RPL node with error (ERN3)	RPL owner (ERN 1) and RPL node (ERN2)	Other Non-RPL node (ERN4)
From the Idle state, ERN3: 1. Blocks the MS port. 2. Sends the RAPS (MS). 3. Flushes the FDB. 4. Enters the manual switch (MS) state.		
		From the Idle state, ERN4: 1. Forward R-APS (MS). 2. Flush the FDB. 3. Enter the MS state.
	From the Idle state, ERN 1: 1. Forwards R-APS (MS). 2. Unblocks the RPL. 3. Flushes the FDB. 4. Enters the MS state.	

After the manual switch is triggered, the operator can clear it with the **no** command and MS recovery will begin. Refer to the following table for the event sequence.

Table 46: MS recovery process event sequence

Non-RPL node with error (ERN3)	RPL owner (ERN1)	RPL node (ERN2) with dual-end blocking enabled	Non-RPL node (ERN4)
From the MS state, ERN3: 1. Stops sending R-APS (MS). 2. Sends R-APS (NR). 3. Continues to block the port. 4. Enters the Pending state.			
	From the MS state, ERN1: 1. Receives the R-APS (NR). 2. Starts the WTB timer. 3. Forwards the R-APS (NR). 4. Enters the Pending state. 5. After the WTB timer expires, blocks the RPL. 6. Flushes the FDB. 7. Sends R-APS (NR, RB). 8. Enters the Idle state.	From the MS state, ERN2: 1. Receives the R-APS (NR). 2. Forwards the R-APS (NR). 3. Enters the Pending state.	From the MS state, ERN2: 1. Receives the R-APS (NR). 2. Forwards the R-APS (NR). 3. Enters the Pending state.
From the Pending state, ERN3: 5. Receives the R-APS (NR, RB) and unblocks the blocking port. 6. Forwards the R-APS (NR, RB). 7. Flushes the FDB. 8. Enters the Idle state.		From the Pending state, ERN2: 4. Blocks the RPL. 5. Forwards the R-APS (NR, RB). 6. Flushes the FDB. 7. Enters the Idle state.	From the Pending state, ERN4: 4. Forwards the R-APS (NR, RB). 5. Flushes the FDB. 6. Enters the Idle state.

Forced Switch

Forced Switch (FS) is an operator-initiated mechanism that moves the blocking role of the RPL to a different ring link followed by unblocking the RPL, even if one or more failed links exist in the ring.

The node configured to initiate an FS blocks the port and sends out a R-APS (FS) to inform other nodes to unblock any blocked ports (including failed ones) as long as no other local request with higher priority exists. The RPL owner unblocks the RPL and flushes the FDB.

Any node accepting a R-APS (FS) message stops transmitting R-APS messages.

Multiple FS instances can be configured in the topology even when the topology is in the same segment where an FS is being cleared by **no** command. When an operator clears an FS on the same node where an FS is configured, this node keeps the port in the blocking state, sends out a R-APS (NR) to adjacent nodes, and starts the guard timer. Other nodes that receive the R-APS (NR) forward the message. When the RPL owner receives this message, the RPL owner starts the WTB timer. When the WTB timer expires, the RPL owner sends out a R-APS (NR, RB), blocks the RPL, and flushes the FDB. Other nodes in the topology that receive the R-APS (NR, RB) unblock any non-failed port and flush the FDB.

An FS request can be accepted no matter what state the topology is in. Since the local FS and R-APS (FS) are higher priority than SF; an SF occur i ng later than FS will not trigger the SF process. In addition, because the local FS and R-APS (FS) are higher priority than SF, when a node receives a R-APS (FS) without any local higher priority event, it will unblock any blocked port. The node with the failed link also unblocks the blocked port; but because the link has failed, the topology is broken into segments.

Since the local FS and R-APS (FS) are higher priority than a local SF clear when the link failure is removed without any local higher priority event, the nodes with the recovering link do not trigger SF recovery.

After the operator clears the FS condition on the node, the node starts the guard timer and sends out a R-APS (NR). When the RPL owner receives a R-APS(NR), it stops the WTB timer and starts the guard timer. The RPL owner blocks the RPL and sends out a R-APS (NR, RB). Any node receiving a R-APS (NR, RB) unblocks the non-failed blocked port. If the guard timer is still running on the node with previous FS, this node ignores R-APS messages until the guard timer expires. The topology is again broken into segments. After this node processes the R-APS (NR, RB), however, it unblocks the blocked node; and the topology is in a loop free state and in one segment.

The following figure shows a port failure on ERN 4.

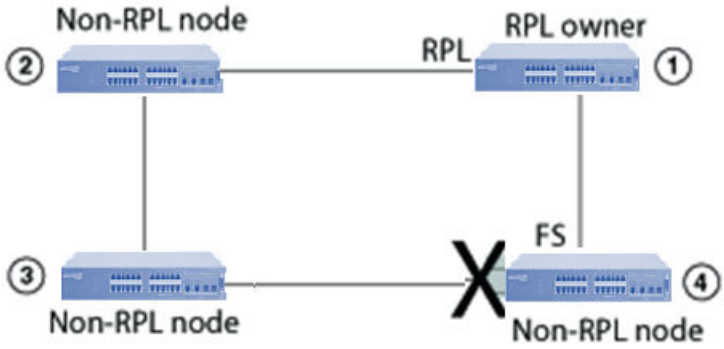


Figure 47: Single forced switch scenario

The following table shows the sequential order of events triggered as a result of an operator-initiated forced switch command entered from ERN 4.

Table 47: Single FS process--operator entered the forced switch command from ERN 4

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
Idle	Idle	Idle	From the Idle state, ERN 4: 1. Processes the Forced Switch command 2. Blocks the requested port 3. Transmits R-APS (FS) 4. Unblocks the non-requested port 5. Flushes the FDB 6. Enters the Forced Switch (FS) state
From the Idle state, ERN 1: 1. Unblocks the RPL 2. Flushes the FDB for first time 3. Forwards R-APS(FS) 4. Enters the FS state	From the Idle state, ERN 2: 1. Unblocks the port 2. Flushes the FDB for the first time 3. Forwards R-APS(FS) 4. Enters the FS state	From the Idle state, ERN 3: 1. Unblocks the port 2. Flushes the FDB for the first time 3. Forwards R-APS(FS) 4. Enters the FS state	
From the FS state, ERN 1 forwards R-APS	From the FS state, ERN 2 forwards R-APS	From the FS state, ERN 3 forwards R-APS	From the FS state, ERN 4: 7. Transmits R-APS(FS) 8. Terminates the received R-APS on the blocking port 9. Terminates its own R-APS(FS)
All ERNs remain in FS state.			

Next, the operator enters the **no** command to clear the forced switch. For this example, the operator initiated the forced switch from ERN 4 and must clear it from ERN 4. The following table shows the forced switch recovery process in sequential order.

Table 48: FS clear process

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
			From the FS state, ERN 4: 1. Starts the guard timer 2. Stops transmitting R-APS(FS 3. Transmits R-APS(NR) 4. Keeps blocking the port 5. Enters Pending state
From FS state, ERN 1: 1. Forwards R-APS 2. Starts the guard timer 3. Starts the WTB timer 4. Enters Pending state			
	From FS state, ERN 2: 1. Forwards R-APS 2. Starts the guard timer 3. Enters the Pending state	From FS state, ERN 3: 1. Forwards R-APS 2. Starts the guard timer 3. Enters the Pending state	
After the WTB timer expires, from the Pending state ERN 1: 5. Blocks the RPL port 6. Transmits R-APS(NR,RB) 7. Unblocks the non-RPL port 8. Flushes the FDB			

Table 48: FS clear process (continued)

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
9. Enters the Idle state			
	From the Pending state, ERN 2: 4. Flushes the FDB 5. Forwards R-APS(NR,RB) 6. Enters the Idle state	From the Pending state, ERN 3: 4. Stops transmitting R-APS 5. Unblocks ports 6. Flushes the FDB 7. Forwards R-APS(NR,RB) Enters the idle state	From the Pending state, ERN 4: 6. Stops transmitting R-APS 7. Unblocks ports 8. Flushes the FDB 9. Forwards R-APS(NR,RB) 10. Enters the Idle state
From the idle state, ERN 1: 10. Receives its own R-APS(NR,RB) 11. Stops transmitting R-APS 12. Remains in the Idle state			

Double Forced Switch

A local FS is of a higher priority than a received R-APS (FS); therefore, the local FS request blocks the port even when the node receives a R-APS(FS) from another FS request of another node.

After the first FS clears, the node starts the guard timer and sends out a R-APS (NR). The adjacent nodes of the first cleared FS node will not process or forward the R-APS (NR) because they are still receiving R-APS (FS) from the second FS node. When the first FS node receives R-APS (FS) from the second FS nodes, it unblocks any blocked port and stops transmitting any lower priority R-APS messages. At this point, the topology follows the single FS process, as previously described.

Dual-end blocking

Dual-end blocking is a user configurable feature to directly conserve bandwidth of the RPL and indirectly conserve processing power of the RPL owner. When you configure a node in a major ring adjacent to the RPL owner to be an RPL node with dual-end blocking enabled, data traffic and R-APS messages will not be forwarded to the blocked port of the RPL owner.

When a failure occurs in the ring and the RPL node (not the RPL owner) receives a R-APS (of type SF, FS, or MS), the RPL node unblocks the configured dual-end blocked port. When the RPL node receives a R-APS (NR, RB), it reblocks the originally

configured dual-end blocked port. To configure dual-end blocking you need to configure the RPL and dual-end blocking on both the RPL owner and the adjacent peer (RPL node).

Non-revertive mode

In non-revertive mode, the traffic channel is allowed to use the RPL, if it is not failed, after a switch condition clears. In the recovery from a Protection state, the RPL owner generates no response regarding the reception of NR messages. When other healthy nodes receive the NR message, there is no action in response to the message. After the operator issues a **no** command for non-revertive mode at the RPL owner, the non-revertive operation is cleared, WTB or WTR timer starts, as appropriate, and the RPL owner blocks its port to the RPL and transmits a R-APS (NR, RB) message. Upon receiving the R-APS (NR, RB), any blocking node should unblock its non-failed port.

Interconnected rings

Interconnected rings consist of one major ring and one or more sub-rings with shared physical links. The ring links between the interconnection nodes are controlled and protected by the ERP ring to which they belong. A sub-ring is similar to the major ring in that each sub-ring has an RPL and an RPL owner. The RPL owner can be configured in any node belonging to the ring.

Refer to the following figure. The dotted lines show two of the many potential sub-rings that you can configure.

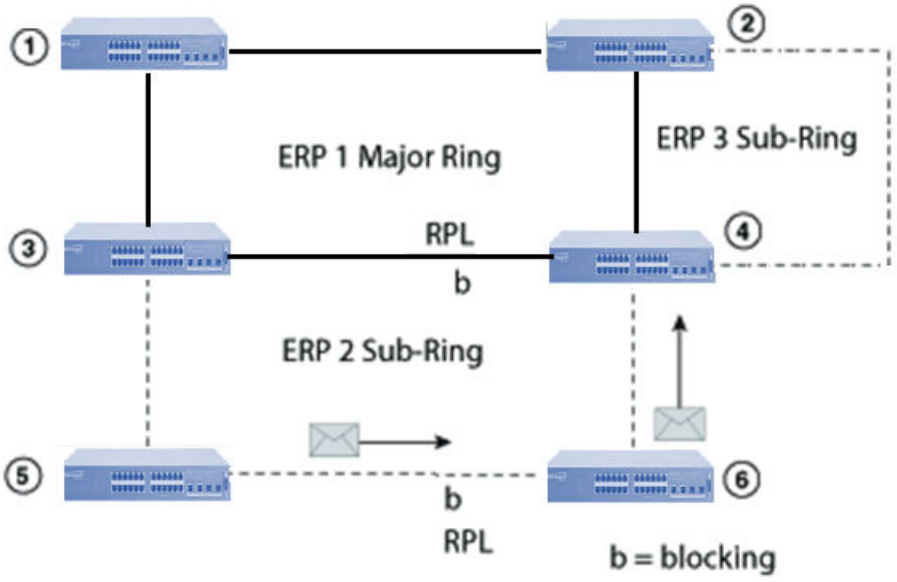


Figure 48: Interconnected rings with major and sub rings shown

When a sub-ring initializes, each ERN in the non-closed ERP sends out a R-APS (NR). After the RPL owner receives a R-APS (NR), it blocks the RPL; and the RPL owner sends out a R-APS (NR, RB). The shared link remains blocked even if the shared link has a SF error. The blocking state in ERP means the R-APS channel is blocked at the same port

where the traffic channel is blocked, except on sub-rings without use of R-APS virtual channel.

Note
ERP Virtual channel support is no longer supported.

A sub-ring in segments interconnecting major rings is not supported. Refer to the following figure, which shows a major ring and two segments not supported as a sub-ring.

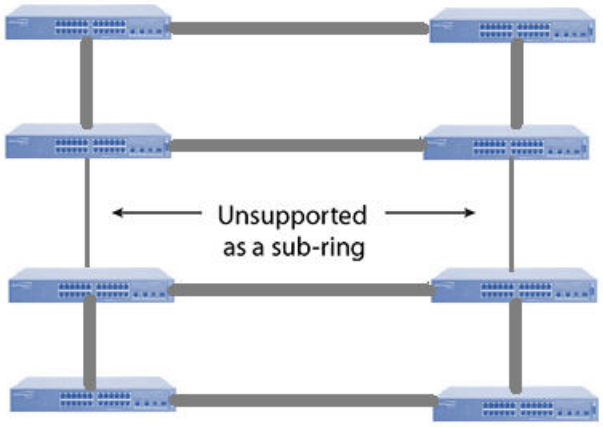


Figure 49: Unsupported sub-ring in segments

Blocking prevents R-APS messages received at one ring port from being forwarded to the other ring port; it does not prevent the R-APS messages locally generated at the ERP control process from being transmitted over both ring ports, and it also allows R-APS messages received at each port to be delivered to the ERP control process.

Each ERN in a major ring terminates R-APS messages received on a blocking port and does not forward the message if the port is in a blocking state. Each ERN in a sub-ring, however, still forwards the R-APS messages received on a blocking port.

Configuring ERP

To configure and initialize ERP using only APS you must set up one RPL owner and one or more Non-RPL nodes. The minimum configuration tasks are listed in this section.

Before configuring ERP, however, you must have already configured a VLAN and ports.

Note
ERP supports topology-groups if the ERP interfaces are in the same VLAN

You must perform the following minimum configuration tasks for the RPL owner:

- Configure an ERP instance.
- Set the left and right interfaces.

- Set the role as owner.
- Set the RPL.
- Enable the configuration.

You must perform the following minimum configuration tasks for each non-RPL node:

- Configure an ERP instance.
- Set the left and right interfaces.
- Enable the configuration.

ERP topology and configuration

Example ERP topology

The following is an example three-node topology for reference in the configuration examples. The topology consists of three devices: an RPL owner, and RPL node, and a non-RPL node. This is a basic configuration. Additional commands can be used to provide additional features.

Do the following before configuring any ERP settings:

- Configure the VLAN.
- Configure the interface as a switchport.
- Add the VLAN to the switchport in either trunk or access mode.
- Configure the **`switchport mode trunk-no-default-native`** command on ports configured as trunk ports.

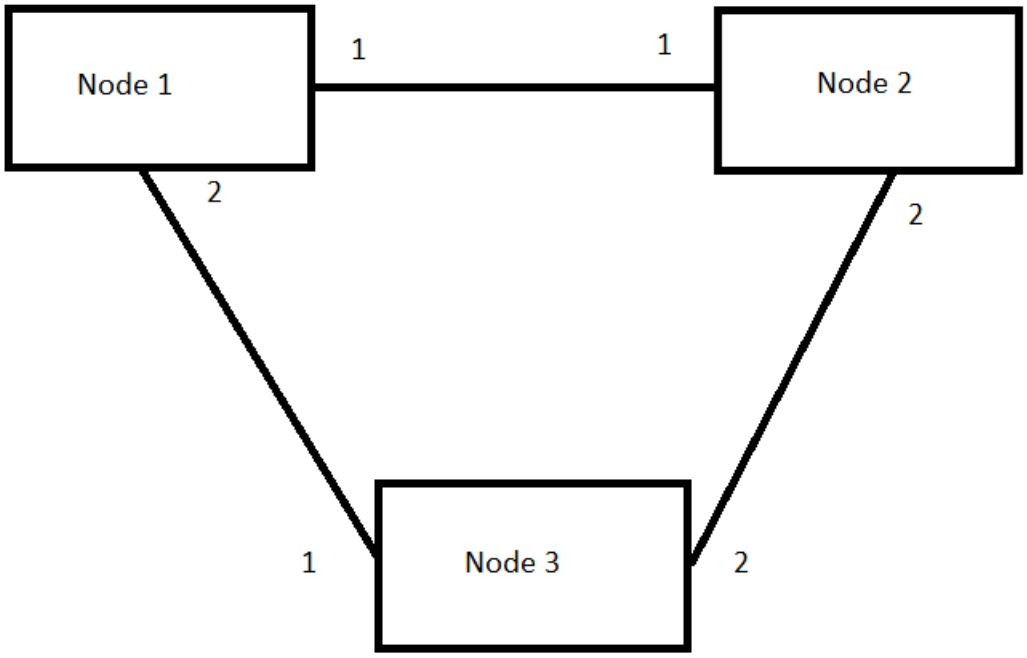


Figure 50: Example ERP topology

ERP configuration examples

The following are example configuration examples on the nodes.

Node 1 (RPL owner)

```
configure terminal
vlan 222
end
configure terminal
interface Ethernet 0/1
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 222
 no shutdown
!
end

configure terminal
interface Ethernet 0/2
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 222
 no switchport trunk tag native-vlan
 no shutdown
!
end

configure terminal
erp 1
 left-interface vlan 222 ethernet 0/1
 right-interface vlan 222 ethernet 0/2
 rpl-owner
```

```
 rpl vlan 222 ethernet 0/1
 enable
!
end
```

Note
Optionally, you can configure the non-revertive node feature. This setting can be configured only on the RPL owner.

Node 2 (RPL node)

```
configure terminal
vlan 222
end

configure terminal
interface Ethernet 0/1
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 222
 no shutdown
!
end

configure terminal
interface Ethernet 0/2
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 222
 no shutdown
!
end

configure terminal
erp 2
 left-interface vlan 222 ethernet 0/2
 right-interface vlan 222 ethernet 0/1
 rpl vlan 222 ethernet 0/1
 enable
!
end
```

Node 3 (Non-RPL node)

```
configure terminal
vlan 222
end

configure terminal
interface Ethernet 0/1
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 222
 no shutdown
!
end

configure terminal
interface Ethernet 0/2
 switchport
 switchport mode trunk-no-default-ntive
 switchport trunk allowed vlan add 222
 no shutdown
!
```

```
end

configure terminal
erp 3
 left-interface vlan 222 ethernet 0/1
 right-interface vlan 222 ethernet 0/2
 enable
!
end
```

Assigning ERP IDs

You must assign an ERP ID, by means of the global **erp** command, to create an ERP instance. This ID number is used to do the following:

- Filter and clear statistics associated with a particular ERP ID
- Delete the non-revertive mode in the case of an RPL owner
- Clear WTR and WTB timers

The *erp_id* value is a number from 1 to 255.

Syntax: **erp** *erp_id*

The following example creates an ERP instance with specified number.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)#
```

Naming an Ethernet Ring Node

The name must be 31 alphanumeric characters or fewer, and can use the "underscore" and "dash" special characters. For example, to assign the name "major-ring-vlan100" to an ERN with ID number 100, enter the following:

```
device# configure terminal
Entering configuration mode terminal
device(config)# erp 100
device(config-erp-100)#
device(config-erp-100)# name "major-ring-vlan100"
```

Use the **no** form of the command to remove the name.

Configuring the default MAC ID

You can configure the MAC ID. The device appends this ID number to the end of the permanent portion of the ERP MAC address in R-APS messages. By default 01-19-A7-00-00-<ERP-ID> is used as the dst MAC, which is always used by Version 1 of ITU-T 8032. If Version 2 is configured, then the **raps-default-mac** command can be negated by entering the **no raps-default-mac** command. The configured ERP ID will appear as the last 8-bit number in the destination MAC.

The following example sets a default R-APS destination MAC address.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# raps-default-mac
```

Configuring a R-APS MEL value

You can configure the R-APS Maintenance Entity Group Level (MEL) value, by means of the **raps-mel** command. This value is carried in ERP PDUs. The default is 7.

The following example sets a nondefault R-APS MEL value.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# raps-mel 6
```

Configuring R-APS topology change propagation

When there is a topology change in a sub-ring, the information needs to be propagated over the major ring. This propagation involves the transmission of R-APS (MAC flush event) PDUs over the major ring associated with the sub-ring. This results in a filter database (FDB) flush on the major ring nodes.You can enable this propagation by means of the **raps-propagate-tc** command on the sub-ring ERP instance.

The following example enables the propagation of topology change information.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# raps-propagate-tc
```

Configuring interfaces

Each Ethernet Ring Node (ERN) in a major ring must have explicitly defined left and right interfaces so that ERP can function correctly.

For proper operation you must configure the interfaces following the same manner on each ERN, such as left/ right, left/ right, and so on.

The following example configures a left and right interface for a major ring.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# right-interface vlan 2 ethernet 1/2
device(config-erp-1)# left-interface vlan 2 ethernet 1/1
```

ERNs in a sub-ring must have at least one interface (either left or right) configured as shown below so that ERP can function correctly. The following example configures a right interface for a sub-ring.

```
device# configure terminal
device(config)# erp 2
device(config-erp-2)# right-interface vlan 2 ethernet 1/1
device(config-erp-2)# sub-ring parent-ring-id 1
```

Enabling the ERP configuration

You must apply the **enable** command to activate an ERP configuration. You can use the **no** command to disable the configuration.

Within an interconnected ring topology, you must first enable ERP on the major ring configured with two interfaces, followed by the sub-ring configured with a single interface. There can be multiple sub-rings configured and activated per single major

ring. However to deactivate or delete ERP on this interconnected ring topology, the sub-ring(s) must be deactivated or deleted first, followed by the major ring.

The following example activates a non-RPL node in a major ring.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# right-interface vlan 2 e 1/2
device(config-erp-1)# left-interface vlan 2 e 1/1
device(config-erp-1)# enable
```

The following example activates ERP for a sub-ring.

```
device# configure terminal
device(config)# erp 2
device(config-erp-2)# right-interface vlan 2 ethernet 1/1
device(config-erp-2)# sub-ring parent-ring-id 1
device(config-erp-2)# enable
```

Assigning the RPL owner role and setting the RPL

Each ring needs to have one RPL owner for each ring. The RPL owner's role is to block traffic on one port when no failure exists in the ring. The blocked port will be the left interface that you initially configured. After configuring the ERN to be the RPL owner, by means of the **rpl-owner** command, you next must set the RPL, by means of the **rpl vlan** command.

The following example illustrates assigning the RPL owner role and setting the RPL.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# rpl-owner
device(config-erp-1)# rpl vlan 5 ethernet 0/1
```

Enabling sub-rings for multi-ring and ladder topologies

In multi-ring and ladder topologies, you can enable the multi-ring feature by means of the **sub-ring** command.

Interconnected rings consist of one major ring and at least one sub-ring within the same VLAN. A sub-ring is not a complete ring. Nodes within a sub-ring can be configured as a one-arm ring. Each sub-ring must have its own RPL owner and RPL ports as appropriate.

RPL ports and the RPL owner also must be configured in a sub-ring. All ERP features are available in both major rings and sub-rings.

R-APS PDUs flow only in the nodes with same ring ID. The R-APS PDU can be forwarded through the port in sub-ring blocking state.

The **parent-ring-id** keyword is used to associate the sub-ring with its parent ring. In such a case, use the parent-ring-id configuration to determine the ring to which the sub-ring is connected. The parent ring can be either another sub-ring or a major ring connected to the sub-ring.

Use the **no sub-ring** command to delete the sub-ring support.

If you have six nodes you can put them in one ring. The latency time for packet transport, however, increases in big topologies even within the same VLAN, so it is better to separate them out.

The following example sets the sub-ring as well as a parent-ring ID of 2.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# sub-ring parent-ring-id 2
```

Configuring non-revertive mode

In **revertive-mode**, once the condition causing a Signal failure has cleared, traffic is blocked on the RPL and restored to the working transport entity. In **non-revertive-mode**, traffic is allowed to use the RPL if it has not failed, even after the device is no longer in Protection state. The link where the failure had occurred continues to remain blocked and the RPL remains unblocked.

After the Ethernet Ring enters a protected state, if you do not want the topology to return to the original state you can use the **non-revertive-mode** command to keep it in the new state. Enter this command on the RPL owner only, and then enter the **enable** command.

The following example configures non-revertive mode and enables the configuration.

```
device# configure terminal
device(config)# erp 100
device(config-erp-100)# non-revertive-mode
```

Use the **no non-revertive-mode** command to remove the non-revertive mode setting.

Configuring and clearing a forced switch

An operator can use the forced switch (FS) mechanism, by means of the **force-switch** command, when no errors, a single error, or multiple errors are present in the topology. You can enter this command multiple times. You need to explicitly specify the VLAN and Ethernet slot and port.

The following example configures and clears a forced switch.

```
device# configure terminal
device(config)# erp 100
device(config-erp-100)# force-switch vlan 100 ethernet 0/10
device(config-erp-100)#
```

Use the **no forced-switch** command to remove the forced switch mechanism.

Configuring and clearing a manual switch

Manual switch (MS) is an operator-initiated process, configured by means of the **manual-switch** command, that manually blocks a desired port in a ring. You need to explicitly specify the VLAN, Ethernet slot, and port from the desired device.

The following example configures and clears a manual switch.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# manual-switch vlan 5 ethernet 0/1
```

Use the **no manual-switch** command to remove the manual switch mechanism.

Configuring the guard timer

The guard timer, configured by means of the **guard-time** command, prevents ERNs from acting upon outdated R-APS messages and prevents the possibility of forming a closed loop. The guard timer enforces a period during which an ERP topology ignores received R-APS.

This timer period should always be greater than the maximum expected forwarding delay in which an R-APS message traverses the entire ring. The longer the period of the guard timer, the longer an ERN is unaware of new or existing relevant requests transmitted from other ERN and, therefore, unable to react to them.

The guard timer is used in every ERN, once a guard timer is started, it expires by itself. While the guard timer is running, any received R-APS request/state and Status information is blocked and not forwarded to the priority logic. When the guard timer is not running, the R-APS request/state and status information is forwarded unchanged.

Note
The ITU-T G.8032 standard defines the guard timer period as configurable in 10 ms increments from 10 ms to 2000 ms (2 seconds) with a default value of 500 ms.

The guard timer is activated when an ERN receives an indication that a local switching request, such as a clear signal fail, manual switch, or forced switch, is cleared.

The guard timer can be configured in 100-ms increments from 1200 ms to 4000 ms (4 seconds); the default value is 1500 ms (1.5 seconds). The guard timer cannot be stopped manually.

The following example configures a guard timer value of 2000 ms.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# guard-time 2000
```

Use the **no guard-time** command to clear the configuration, restoring it to the default value.

Configuring and clearing the WTR timer

For Signal Fail (SF) recovery situations, you can use the **wtr-time** command to configure the Wait-To-Restore (WTR) timer on the RPL owner to prevent frequent operation of the protection switching due to the detection of intermittent signal failures. When recovering from a Signal Failure, the WTR timer must be long enough to allow the recovering network to become stable.

This WTR timer is activated on the RPL Owner Node. When the relevant delay timer expires, the RPL owner initiates the reversion process by transmitting an R-APS (NR, RB) message. The WTR timer is deactivated when any higher-priority request preempts this timer. The WTR timers may be started and stopped. A request to start running the WTR timer does not restart the WTR timer. A request to stop the WTR timer stops the WTR timer and resets its value. The **clear erp wtr-time** command can be used to stop the WTR timer. While the WTR timer is running, the WTR running signal is continuously generated. After the WTR timer expires, the WTR running signal is stopped, and the WTR Expires signal is generated. When the WTR timer is stopped by the **clear erp wtr-time** command, the WTR Expires signal is not generated.

When configured, the RPL owner waits until the timer expires before transmitting the R-APS (NR, RB) message to initiate the reversion process. While the timer is in effect, the WTR running signal is continuously generated. You can configure the WTR timer in 1 minute increments from 1 to 12 minutes; the default value is 5 minutes.

The following example configures a WTR time of one minute.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# wtr-time 1
```

Use the **no wrt-time** command to clear the configuration in ERP configuration mode, restoring it to the default value.

Use the global **clear erp wtr-time** command to clear the WTR timer for a specific ERP instance, as in the following example for instance 1.

```
device# clear erp wtr-time 1
```

Testing the WTR timer

Use the **fast-wtr-time** command to change the timer's unit of measure from minutes to seconds, allowing you to test your configuration sooner. Instead of having to wait 5 minutes for the timer to expire, you wait 5 seconds.

The following example configures a WTR time value to seconds.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# fast-wtr-time
```

Use the **no fast-wtr-time** command to return the unit of measure to minutes.

Configuring and clearing the WTB timer

The Wait To Block (WTB) timer ensures that clearing of a single Forced Switch (FS) command does not trigger the reblocking of the RPL when multiple FS situations co-exist in an Ethernet Ring. When recovering from a Manual Switch (MS) or FS command, the delay timer must be long enough to receive any latent remote FS or MS.

While it is running, the WTB running signal is continuously generated. The WTB timer is 5000 ms (5 seconds) longer than the guard timer. You can configure this timer in 100-ms increments from 5100 to 7000 ms (7 seconds); the default value is 5500 ms.

The WTB timer can be stopped by means of the **clear erp wtb-time** command.

The following example configures a WTB time of 5100 ms.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# wtb-time 5100
```

Use the **no wrb-time** command to clear the configuration in ERP configuration mode, restoring it to the default value.

Use the global **clear erp wtb-time** command to clear the WTB timer for a specific ERP instance, as in the following example for instance 1.

```
device# clear erp wtb-time 1
```

Configuring a hold-off timer

The hold-off timer, configured by means of the **holdoff-time** command, is used in each ERN to prevent unnecessary Signal Fail (SF) events caused by port flapping. If you configure a non-zero hold-off timer value, when a link error occurs, the event is not reported immediately. When the hold-off timer expires, ERP checks to see whether the error still exists.

The hold-off timer is used in every ERN. When a new defect occurs (new SF), this event is not reported immediately to trigger protection switching if the provisioned hold-off timer value is non-zero. Instead, the hold-off timer is started. When the hold-off timer expires, the trail that started the timer is checked as to whether a defect still exists. If one does exist, that defect is reported and protection switching is triggered.

You can configure the hold-off timer in 100-ms increments from 0 to 10,000 ms (10 seconds); the default value is 0 ms. The hold-off timer value cannot be stopped through the CLI.

The following example configures a holdoff-time of 100 ms.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# holdoff-time 100
```

Use the **no holdoff-time** command to clear the configuration, restoring it to the default value.

The message interval time of R-APS messages continuously sent within an ERP ring can be configured by means of the **message-interval** command. You can configure the interval in 100-ms increments from 100 to 5000 ms (5 seconds); the default value is 5000 ms.

The following example configures a message interval of 100 ms.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# message-interval 100
```

Use the **no message-interval** command to clear the configuration, restoring it to the default value.

Setting the ITU-T G.8032 version number

You can configure the ERP configuration to use G.8032 version 1 or 2. The default value is version 2.

The following example configures Version 1.

```
device# configure terminal
device(config)# erp 1
device(config-erp-1)# version 1
```

Use the **no version** command to clear the configuration, restoring it to the default value.

You can view the version by entering the **show erp** command. The version appears on the top line directly after the ERP ID.

The following example displays information about all ERP instances, including the version.

```
device# show erp
ERP 5 (Version 2) - VLAN 6
================================================================================
Erp ID  Status     Oper              Node          Non-revertive      Topo
                   state             role          mode               group
5       enabled    Idle              rpl-node      disabled           -

Fast         Ring         WTR        WTB        Guard      Holdoff    Msg
convergence  type         time(min)  time(ms)   time(ms)   time(ms)   intv(ms)
enabled      Major ring   5          5500       1500       0          5000

Raps-default-mac   Parent-ring-erp-id   Raps-propagate-tc   Mel-Config   Mel-Oper
ON                 0                    OFF                 2            3

I/F     Port       ERP port state   Interface status   Interface type
L       eth0/4     forwarding       normal              non-rpl
R       eth0/7     blocking         normal              rpl
```

Configuring ERP with IEEE 802.1ag

To configure and initialize ERP using APS and IEEE 802.1ag (Dot1ag), you must set up one RPL owner and one or more non-RPL nodes. Other nonparticipating switches can exist in the ring.

You must perform the following minimum configuration tasks for the RPL owner:

- Configure an ERP instance.
- Set the left and right interfaces.
- Set the role as owner.
- Set the RPL.
- Enable the configuration.

You must perform the following minimum configuration tasks for each non-RPL node:

- Configure an ERP instance.
- Set the left and right interfaces.
- Configure the maintenance entity group end points (MEPs) from each ERN, which can have a role of RPL owner or non-RPL node, adjacent to switches not participating in the ERP configuration.
- Enable the configuration.

IEEE 802.1ag can be used to monitor the ERP interfaces for signal failures. The **dot1ag-compliance** command allows MDs and MAs configured as part of IEEE 802.1ag to be associated with an ERP instance. Note the following:

- With Dot1ag compliance enabled, ERP relies on signal fail or signal OK messages from the Dot1ag module.
- Before enabling Dot1ag compliance in ERP, the user must create a CFM session between the links of the ERP instance.
- For each ring interface, an MEP must be configured and the direction of the MEP must be down.
- Once the CFM session is configured, the user must associate an MEP with the corresponding ERP ring interface.
- With a CCM interval of 3.3 ms, CFM can detect a link failure within 10 ms. This helps ERP to achieve faster protection switching times, with traffic converging within 50 ms.

The following example configures IEEE 802.1ag compliance.

```
device(config)# erp 1
device(config-erp-1)# dot1ag-compliance
device(config-dot1ag-compliance)# left-interface domain-name md1 ma-name ma1 mep 2 remote-
mep 4
device(config-dot1ag-compliance)# right-interface domain-name md1 ma-name ma3 mep 2
remote-mep 4
device(config-dot1ag-compliance)# enable
```

Use the **no dot1ag-compliance** command to disable the feature.

The **domain-name** parameter specifies the MD name for 802.1ag CFM.

The **ma-name** parameter specifies the maintenance association name. This can be up to 21 characters long.

Refering to the example topology above, corresponding configurations for this feature on Node 1 and Node 2 are as follows.

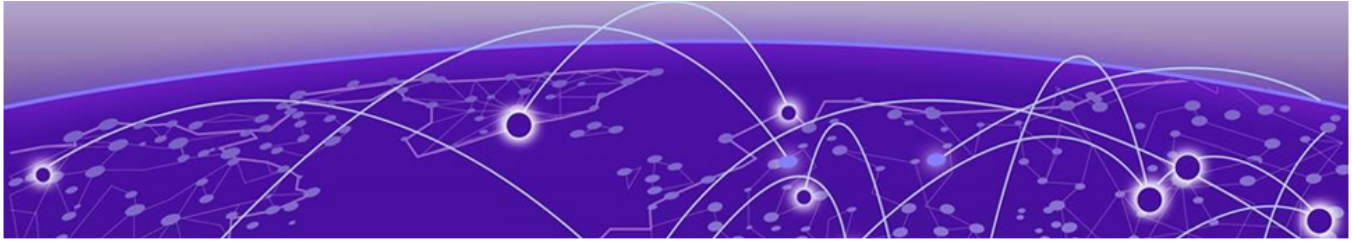
Node 1 Dot1ag configuration

```
protocol cfm
domain-name md1 id 1 level 7
ma-name ma2 id 2 vlan 222 priority 7
ccm-interval 3.3-ms
mep 2 down ethernet 0/1
!
!
ma-name ma3 id 3 vlan 222 priority 7
```

```
ccm-interval 3.3-ms
mep 1 down ethernet 0/2
!
erp 222
left-interface vlan 222 ethernet 0/1
right-interface vlan 222 ethernet 0/2
dot1ag-compliance
left-interface domain-name md1 ma-name ma2 mep 2 remote-mep 1
right-interface domain-name md1 ma-name ma3 mep 1 remote-mep 2
enable
end
```

Node 2 Dot1ag configuration

```
protocol cfm
domain-name md1 id 1 level 7
  ma-name ma2 id 2 vlan 222 priority 7
   ccm-interval 3.3-ms
   mep 1 down ethernet 0/1
   !
  !
  ma-name ma1 id 1 vlan 222 priority 7
   ccm-interval 3.3-ms
   mep 2 down ethernet 0/2
!
erp 222
left-interface vlan 222 ethernet 0/1
right-interface vlan 222 ethernet 0/2
dot1ag-compliance
  left-interface domain-name md1 ma-name ma2 mep 1 remote-mep 2
  right-interface domain-name md1 ma-name ma1 mep 2 remote-mep 1
!
rpl-owner
rpl vlan 222 ethernet 0/2
enable
end
```



ETH-CSF

ETH-CSF overview on page 313
Configuring ETH-CSF on page 319

ETH-CSF overview

Note
For text of the standard, see https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.8013-201508-I!!PDF-E&type=items

This feature is used by a Maintenance Entity Group End Point (MEP) to propagate the detection of a failure or defect event in an Ethernet client signal to a peer Remote MEP (RMEP) when the client itself does not support appropriate fault detection, defect detection, or propagation mechanisms such as Ethernet Continuity Check (ETH-CC) or Ethernet Alarm Indication Signal (ETH-AIS). The ETH-CSF messages propagate in the direction from the Ethernet MEP, associated with the ingress client port detecting the failure or defect event, to the Ethernet peer RMEP.

Note
For supporting documentation, refer to "Y.1731 Performance Monitoring" in the "Operation, Administration, and Maintenance" chapter in the *Extreme SLX-OS Monitoring Configuration Guide*.

ETH-CSF use case

The NE device could be connected to another NE device of a core network of another ISP or its own core network on a different geographical location through an international link on a WAN. The NE device on the remote side would connect to the remote PE device to which the remote CE devices are connected.

The two PE devices that form an Ethernet virtual circuit (EVC) are connected through a Down MEP link. Now the CE devices on both local and remote sites form the client for the respective PE devices, and the connecting ports on these PE devices to the respective CE devices on local and remote sites are the designated Ethernet client ports. These client ports can be either physical interfaces or port-channel interfaces. Only one client port can be associated with an MEP. Thus, the scale for the client ports that can be associated to MEPs with a one-to-one mapping is limited by the number of MEPs that can be configured (currently 8 k) on the device, a number that is large

enough compared to the total number of physical interfaces available on the device. Hence there is practically no limit to the number of ETH-CSF client ports that can be created on a device.

Note
For a discussion of MEP and other connectivity fault management (CFM) issues, see the chapter "802.1ag Connectivity Fault Management" in the *Extreme SLX-OS Layer 2 Switching Configuration Guide*.

The purpose of the feature is to propagate the ETH-CSF indication from an MEP configured on the PE device to the remote MEP on the peer PE device when a port on CE device connected to this PE device goes down. Thus, the MEP on the PE device detects a link failure event on its ingress Ethernet client port and propagates this client signal fault to the RMEP on the remote peer PE side. The following figure illustrates an example topology.

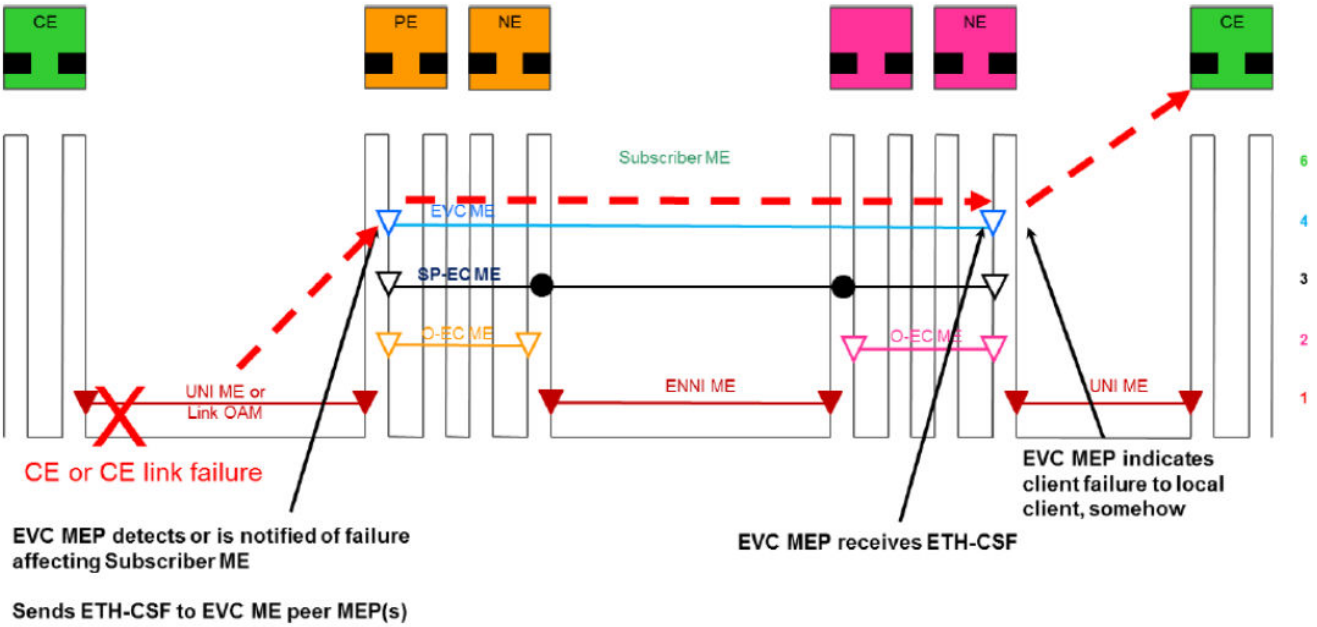


Figure 51: CSF example topology

Upon reception of the CSF indication at the RMEP on the peer side, this fault signal is further propagated to the egress Ethernet client port by operationally bringing the port down. Because the CE device is connected directly to the remote PE device through a point-to-point link, the remote CE device detects an Ethernet link failure on its connected port, and the admin can take necessary actions immediately to restore the connectivity between the CE devices.

This feature is specifically helpful when the client itself (the CE device) does not support any means of notification to its peer (the remote CE device), such as through ETH-AIS or the RDI function of ETH-CC.

The following are the requirements for this feature according to the MEF 30.1 standard:

- ETH-CSF transmissions SHOULD be disabled on an MEP by default.
- ETH-CSF transmissions SHOULD be enabled only on MEPs in point-to-point MEGs.
- Transmission periods of 1 second and 1 minute MUST be supported for ETH-CSF.
- The ETH-CSF default transmission period SHOULD be 1 second.

ETH-CSF specifications

ETH-CSF is applicable only to point-to-point Ethernet transport applications, which means that it can work only with Down MEPs.

The following specific configuration information is required by an MEP to support ETH-CSF transmission:

- Local MEG (or MD) level: MEG (or MD) level at which the initiating MEP operates
- ETH-CSF transmission period: Determines transmission periodicity of frames with ETH-CSF information
- Priority: Identifies the priority of frames with ETH-CSF information
- Drop eligibility: Frames with ETH-CSF information are always marked as drop ineligible

The following specific configuration information is required by an MEP to support ETH-CSF reception:

- Local MEG (or MD) level: MEG (or MD) level at which the receiving MEP operates

Note
An MIP is transparent to frames with ETH-CSF information and therefore does not require any information to support ETH-CSF functionality.

The ETH-CSF message indicates also the type of defect. Three CSF defect types are currently defined:

- Client Loss of Signal (C-LOS)
- Client Forward Defect Indication (C-FDI)
- Client Reverse Defect Indication (C-RDI)

The PDU used to convey ETH-CSF information is referred to as the CSF PDU. Frames carrying the ETH-CSF indications are also referred to as CSF frames.

ETH-CSF and port-channel

In such a case, the port-channel interface on a PE device must be associated with the down MEP, which when detects the port-channel link down event. That event triggers a fault indication to be propagated to the peer RMEP, which will then bring down the port-channel interface associated with it on the remote side.

Also, the international link or the WAN between the connected NE devices could been deployed with port-channel links. In this case, it is expected that this port-channel link

carries the ETH-CSF information towards the remote side as does a regular Ethernet link.

CSF transmission

As a result, the frames with ETH-CSF information are not propagated until and unless there is a fault detected on the corresponding ingress client port. An appropriate log message is thrown to indicate the fault detected at the Ethernet client port and the transmission of the ETH-CSF message to the RMEP.

The transmission of packets with CSF information can be enabled or disabled on an MEP through a CLI configuration.

Upon receiving an Ethernet CSF notification from the ingress client port, the associated MEP can immediately start the periodic transmission of frames with ETH-CSF information. This continues until the Ethernet CSF indication is cleared by the source MEP when the local fault with the associated client port is cleared, that is, when it witnesses the link up event in this case.

The clearance of the Ethernet CSF condition by the source MEP can be communicated to the peer RMEP to clear its Ethernet client fault condition in one of two ways:

- It ceases sending ETH-CSF frames for the peer to timeout.
- It propagates an ETH-CSF PDU with Client Defect Clear Indication (C-DCI) information.

CSF reception

This received Ethernet C-DCI information is further propagated by the RMEP to the corresponding egress client port when the Ethernet client port is either operationally shut or enabled, respectively, so that the Ethernet client would detect a link down or link up event, respectively.

An Ethernet MEP detects an Ethernet remote CSF condition when an ETH-CSF PDU with no C-DCI information is received.

The clearance of the Ethernet remote CSF condition by the Ethernet client of a RMEP is detected when either of the following occurs:

- No ETH-CSF frame is received within an interval of 3.5 times (as recommended by the standard) the CSF transmission period on the source MEP in milliseconds.
- An ETH-CSF PDU with C-DCI information is received.

Upon clearance of the CSF condition by the RMEP, the corresponding Ethernet client port is enabled, and the Ethernet client device would detect the link up event for the port towards the network.

An appropriate log message is thrown to indicate either receiving or clearing of the fault indication (by receiving C-DCI information or as the result of a timeout), with subsequent action on the client port by the RMEP.

Note
In case of an RMEP failure that is due to a Continuity Check Message (CCM) not being received, the client port at the remote side associated with such an RMEP is brought up and made operational.

ETH-CSF PDU structure

The format of the CSF PDU used to support the ETH-CSF function is shown in the following figure.

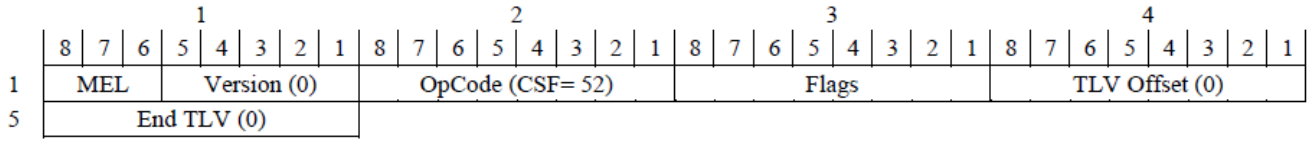


Figure 52: CSF PDU format

The fields of the CSF PDU format are described in the following table.

Table 49: CSF PDU format fields

Field	Description
MEG Level	3-bit field to carry the local MEG level
Version	0
OpCode	PDU type is CSF (52)
Flags	3-bit Type sub-element and a 3-bit Period sub-element

The format of the Flags field is illustrated in the following figure.

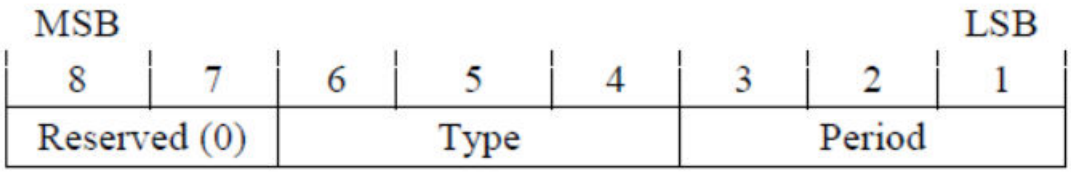


Figure 53: Flags format

The Type field uses bits 6 through 4 to indicate the CSF type, with values as shown in the following table.

Table 50: CSF Type values

Flags [6:4]	Type	Description
000	LOS	Client loss of signal
001	FDI/AIS	Client forward defect indication
010	RDI	Client reverse defect indication
011	DCI	Client defect clear indication

The Period field uses bits 3 through 1 to indicate the transmission period, with values as shown in the following table.

Table 51: CSF Period values

Flags [3:1]	Type	Description
000	Invalid value	Invalid value for CSF PDUs
001	For further study	For further study
010	For further study	For further study
011	For further study	For further study
100	1 second	1 frame per second
101	For further study	For further study
110	1 minute	1 frame per minute
111	For further study	For further study

TLV Offset is set to 0.

End TLV is an all-zeroes octet value.

ETH-CSF considerations and limitations

- Hitless failover is not supported.
- ETH-CSF can be supported only with Down MEPs for point-to-point Ethernet applications. MEPs should be in the Up state and not in a faulty or failed state.
- ETH-CSF frames are transmitted only when the Down MEP (both source and remote) is in the Up state and the transmissions cease case of a MEP failure.
- A client port that is associated with a Down MEP can be either a physical interface or a port-channel interface. No other type of interface is supported.
- There can be only one physical or port-channel interface that is associated with an Down MEP. It is not allowed to associate a physical interface that is already a port-

channel member as a client port to an Down MEP. No two physical or port-channel interfaces can be associated with a single Down MEP.

- All available physical interfaces on a device can be associated as client ports to different Down MEPs with a one-to-one mapping.
- When a client interface is configured with a port-channel interface, the timeout before RMEP sends C-LOS messages back to the MEP when its client interface link is down is 3.5 times the configured tx-interval.

Note
Because 8 k MEPs can be created on the device, practically all of the available client ports can be associated with different MEPs with a one-to-one mapping. Therefore, the scale is the same as that for the number of MEPs that can be created.

Configuring ETH-CSF

Note
Refer to the "802.1ag Connectivity Fault Management" chapter in the *Extreme SLX-OS Layer 2 Switching Configuration Guide*. For command details, see the *Extreme SLX-OS Command Reference*.

1. Configure ETH-CSF on a physical interface.
   a. Enter global configuration mode.
   ```
   device# configure terminal
   ```
   b. Enter the **protocol cfm** command to enter Connectivity Fault Management (CFM) protocol configuration mode.
   ```
   device(config)# protocol cfm
   ```
   c. In CFM protocol configuration mode, enter the **domain-name** command to specify a Maintenance Domain, ID, and level.
   ```
   device(config-cfm)# domain-name md1 id 1 level 3
   ```
   d. Enter the **ma-name** command to specify an MA name, ID, VLAN, and priority.
   ```
   device(config-cfm-md-md1)# ma-name ma1 id 1 vlan 10 priority 7
   ```
   e. Enter the **mep** command to specify an MEP ID, state, and interface.
   ```
   device(config-cfm-ma-ma1)# mep 1 down ethernet 1/1
   ```
   f. Enter the **client-interface** command to specify a client interface, CSF type, and transmission period.
   ```
   device(config-cfm-ma-mep-1)# client-interface ethernet 1/1 csf-type loss-of-signal
   tx-period 1-minute
   ```
2. To configure ETH-CSF on a port-channel interface, configure it as follows.
```
device# configure terminal
Entering configuration mode terminal
device(config)# protocol cfm
device(config-cfm)# domain-name md1 id 1 level 3
device(config-cfm-md-md1)# ma-name ma1 id 1 vlan 10 priority 7
device(config-cfm-md-ma-ma1)# mep 1 down port-channel 100
device(config-cfm-md-ma-mep-1)# client-interface port-channel 200 csf-type loss-of-
signal tx-period 1-minute
```

3. Enter the **show cfm y1731 client-signal-fail** command to verify the configuration.

```
device# show cfm y1731 client-signal-fail
-------------------------------------------------------------------------------
Domain Name : md1
MA Name     : ma1
-------------------------------------------------------------------------------
ETH-CSF Statistics :
-------------------------------------------------------------------------------
MEP RMEP MEP    RMEP   Client CSF    Transmit  Transmit Receive  Transmit Receive
ID  ID   Status Status I/F    Type   Period    Frames   Frames   C-DCI    C-DCI
-------------------------------------------------------------------------------
1   2    UP     UP     1/1    C-LOS  1 minute  0        0        0        0
2   1    UP     DOWN   1/2    C-LOS  1 second  0        0        0        0
```

4. Enter the **clear cfm y1731 client-signal-fail statistics** command to clear the statistics.

```
device# clear cfm y1731 client-signal-fail statistics
```

This example displays ETH-CSF configuration information and packet statistics after the **clear cfm y1731 client-signal-fail** command is issued.

```
device# show cfm y1731 client-signal-fail
-------------------------------------------------------------------------------
Domain Name : md1
MA Name     : ma1
-------------------------------------------------------------------------------
ETH-CSF Statistics :
-------------------------------------------------------------------------------
MEP RMEP MEP    RMEP   Client CSF    Transmit  Transmit Receive  Transmit Receive
ID  ID   Status Status I/F    Type   Period    Frames   Frames   C-DCI    C-DCI
-------------------------------------------------------------------------------
1   2    UP     UP     1/1    C-LOS  1 minute  0        0        0        0
2   1    UP     DOWN   1/2    C-LOS  1 second  0        0        0        0
```